



TUGAS AKHIR - KI141502

PERANCANGAN DAN DESAIN CRM PADA APLIKASI CALLTENANT

**KANIA AMALIA
NRP 5114100172**

Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Dosen Pembimbing II
Abdul Munif, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

RBTC



TUGAS AKHIR - KI141502

PERANCANGAN DAN DESAIN CRM PADA APLIKASI CALLTENANT

**KANIA AMALIA
NRP 5114100172**

**Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.**

**Dosen Pembimbing II
Abdul Munif, S.Kom., M.Sc.**

**Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

(Halaman ini sengaja dikosongkan)

RBTC



UNDERGRADUATE THESES - KI141502

CRM DESIGN FOR CALLTENANT APPLICATION

KANIA AMALIA
NRP 5113100172

First Advisor
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Second Advisor
Abdul Munif, S.Kom., M.Sc.

Department of Informatics
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya 2018

LEMBAR PENGESAHAN
PERANCANGAN DAN DESAIN CRM PADA
APLIKASI CALLTENANT

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

KANIA AMALIA
NRP. 5114 100 172

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.tech.Ir. Raden Venantius Hari Gungardjo
M.Sc.
NIP. 196505181992031003
(Pembimbing I)
2. Abdul Munif, S.Kom., M.Sc.
NIP. 198608232015041004
(Pembimbing II)

SURABAYA
MARET, 2018

(Halaman ini sengaja dikosongkan)

RBTC

PERANCANGAN DAN DESAIN CRM PADA APLIKASI CALLTENANT

Nama Mahasiswa : Kania Amalia
NRP : 5114100172
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1: Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Dosen Pembimbing 2: Abdul Munif, S.Kom., M.Sc.

Abstrak

Tujuan dari penelitian ini adalah untuk menghasilkan CallTenant yang merupakan aplikasi yang menghubungkan antara event organizer dan tenant dengan menerapkan Customer Relationship Management sebagai acuan aplikasi CallTenant. CRM bertujuan untuk mengubah pengelolaan data dan pelayanan terhadap pelanggan yang masih manual serta mencegah perusahaan dari kerugian serta kurangnya strategi untuk mendapatkan dan mempertahankan pelanggan. Peneliti menggunakan metode deskriptif kualitatif dengan mengobservasi kondisi sistem dan pasar yang sudah ada sebagai objek penelitian. Dari hasil penelitian tersebut maka diperoleh data dan bukti bahwa CRM dibutuhkan pengguna untuk mencari suatu informasi dengan cepat, pengguna dapat mendapatkan pelanggan baru, mempertahankan pelanggan, serta meningkatkan hubungan antar event dan tenant dalam kasus CallTenant ini. Sebagai tambahan bagi perusahaan, CRM dapat mengurangi biaya operasional karena aplikasi dapat diakses kapan pun dan di mana pun, meningkatkan produktivitas kerja, meningkatkan jumlah pelanggan, dan meningkatkan tingkat kepuasan pelanggan. Hasil akhir dari penelitian ini adalah menghasilkan CRM yang akan diterapkan pada CallTenant untuk memfasilitasi event dan tenant dalam menjalankan aktivitasnya. CRM memiliki tiga tahap penting yaitu 'Acquire', 'Enhance', dan 'Retain' yang digunakan sebagai acuan utama dalam mendirikan perusahaan.

Kata kunci: CallTenant, CRM, Pelanggan

RBTC

CRM DESIGN FOR CALLTENANT APPLICATION

Student's Name : Kania Amalia
Student's ID : 5114100172
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Second Advisor : Abdul Munif, S.Kom., M.Sc.

Abstract

The purpose of this research is to produce CallTenant application which is an application that connects between event organizer with tenants by applying Costumer Relationship Management as a reference of the application. CRM aims to address the issues of dealings and service to customers that are still conducted manually and prevent companies from losing due to lack of corporate strategy to acquire and retain customers. The researcher uses qualitative descriptive method that is by observing the condition of the system and market condition as the object of research. From the research results obtained data and facts that CRM application is required for users such as looking for information quickly, acquire new customers, improve the relationship between events and tenants, and retain customers. In addition, for the company, this system reduces operational costs, application can be accessed anytime and anywhere, increase work productivity, increase the number of customers, and increase customer satisfaction. The result of this research is to produce CRM application which applied to CallTenant to facilitate event organizer and tenant in running its activities. CRM which is also called Customer Relationship Management has three important stages, "Acquire", "Enhance", and "Retain" which in the future is used as a reference to the founding of company.

Keyword: CallTenant , CRM, Customer

(Halaman ini sengaja dikosongkan)

RBTC

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

“Perancangan dan Desain CRM pada Aplikasi CallTenant”

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Eddy Poerwanto, Ibu Hera Artati, Kakak Zurie Karina, Kakak Tara Indira, Kakak Susiyanti, dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Dr.Tech Ir. R.V. Hari Ginardi, M.Sc dan Bapak Abdul Munif S.Kom., M. Sc. selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.

5. Muhammad Hilman, Tiara Anggita, Rahmatin Nadia, Aviananda Dwirahma, Anindita Larasati, Anne Annisa, Vinsensia Sipriana, Emmanuella Isabella, Roida Pitta, Nadya Pitta, Cecilia Doviandra, Ayu Dewi, Indira Larasati, Ozsadevi Sonia, Aloysia Monika, Felizia Bela yang selama ini memberikan semangat dalam hal studi maupun masalah lainnya.
6. Teman-teman Administrator Lab Manajemen Informasi (MI).
7. Teman-teman angkatan 2014 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama kurang lebih 3,5 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan Tugas Akhir ini.
8. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Desember 2017

Kania Amalia

DAFTAR ISI

LEMBAR PENGESAHAN	
Abstrak	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL	xxvii
DAFTAR KODE PROGRAM	xxix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi.....	3
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur.....	3
1.6.3 Analisis dan Desain Perangkat Lunak	4
1.6.4 Implementasi Perangkat Lunak	4
1.6.5 Pengujian dan Evaluasi	4
1.6.6 Penyusunan Buku	4
1.7 Sistematika Penulisan Laporan	4
BAB II DASAR TEORI.....	7
2.1 Aplikasi Sejenis	7
2.1.1 PusatBazaar.com	7
2.2 CRM Sejenis	8
2.2.1 ZohoCRM.....	8
2.2.2 Perbandingan CRM CallTenant dengan CRM Sejenis	8
2.3 <i>Event Organizer</i>	9
2.4 <i>Tenant</i>	10
2.5 CRM (<i>Customer Relationship Management</i>)	10
2.5.1 Konsep CRM.....	10
2.5.2 Teknologi CRM	11
2.5.3 Tipe CRM.....	11
2.5.4 Tahapan CRM	12
2.5.5 Implementasi CRM pada Saat Ini.....	12
2.6 <i>Customer Lifecycle</i> (Siklus Hidup Pelanggan)	13

2.7	<i>Customer Retention</i> (Retensi Pelanggan)	14
2.8	Loyalitas Pelanggan	15
2.9	<i>Wallet</i>	16
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		19
3.1	Analisis Sistem	19
3.1.1	Deskripsi Umum	19
3.1.2	Deskripsi Implementasi CRM	19
3.1.3	Tahapan CRM untuk Sistem	22
3.1.3.1	Tahap Acquire (Memperoleh Pelanggan Baru).....	22
3.1.3.2	Tahap Enhance (Meningkatkan Hubungan dengan Pelanggan)	23
3.1.3.3	Tahap Retain (Mempertahankan Pelanggan)	24
3.1.4	Analisis Proses Bisnis	24
3.1.5	Analisis Kebutuhan Sistem	27
3.1.6	Analisis Aktor	31
3.2	Perancangan Sistem.....	33
3.2.1	Kasus Penggunaan	33
3.2.1.1	Mengelola Data <i>Frequently Asked Questions</i>	34
3.2.1.2	Mengelola Pelaporan Keuangan.....	39
3.2.1.3	Membalas Pesan <i>Contact Us</i>	47
3.2.1.4	Mengelola Data <i>Reward dan Point</i>	49
3.2.1.5	Mengelola Pelaporan Data <i>Tenant</i>	58
3.2.1.6	Mengelola Pelaporan Data <i>Event</i>	61
3.2.1.7	Mengelola Pelaporan Data <i>User</i>	64
3.2.1.8	Melihat Hasil <i>Feedback</i>	65
3.2.1.9	Menggunakan <i>Reward Event</i>	67
3.2.1.10	Menggunakan <i>Reward Tenant</i>	69
3.2.1.11	<i>Event</i> Mendapatkan <i>Point</i>	71
3.2.1.12	<i>Tenant</i> Mendapatkan <i>Point</i>	73
3.2.1.13	Mengisi <i>Feedback</i>	75
3.2.1.14	Mengisi <i>Wallet</i>	76
3.2.1.15	Melihat Riwayat Konfirmasi <i>Top-up Wallet</i>	79
3.2.1.16	Melihat Riwayat Pesan <i>Contact Us</i>	80
3.2.1.17	Melihat Riwayat Pemakaian <i>Point</i>	81
3.2.1.18	Melihat Riwayat Pemakaian <i>Wallet</i>	82
3.2.1.19	Mengelola Data <i>Review Tenant</i>	84
3.2.1.20	Mengelola Data <i>Review Event</i>	86
3.2.1.21	Menggunakan Fitur <i>Contact Us</i>	88

3.2.1.22	Menggunakan Fitur <i>Frequently Asked Questions</i> ..	89
3.2.1.23	Mengisi <i>Review</i> untuk <i>Tenant</i>	91
3.2.1.24	Mengisi <i>Review</i> untuk <i>Event</i>	93
3.2.1.25	Mendaftar sebagai <i>Event</i> atau <i>Tenant</i>	95
3.2.2	Perancangan Basis Data.....	97
3.2.3	Perancangan Antarmuka.....	97
3.2.3.1	Rancangan Antarmuka Mengelola Data <i>Frequently Asked Questions</i>	97
3.2.3.2	Rancangan Antarmuka Mengelola Pelaporan Keuangan	99
3.2.3.3	Rancangan Antarmuka Membalas Pesan <i>Contact Us</i>	103
3.2.3.4	Rancangan Antarmuka Mengelola Data <i>Reward</i> dan <i>Point</i>	104
3.2.3.5	Rancangan Antarmuka Mengelola Pelaporan Data <i>Tenant</i>	107
3.2.3.6	Rancangan Antarmuka Mengelola Pelaporan Data <i>Event</i>	108
3.2.3.7	Rancangan Antarmuka Mengelola Pelaporan Data <i>User</i>	110
3.2.3.8	Rancangan Antarmuka Melihat Hasil <i>Feedback</i> ..	110
3.2.3.9	Rancangan Antarmuka Menggunakan <i>Reward Event</i>	111
3.2.3.10	Rancangan Antarmuka Menggunakan <i>Reward Tenant</i>	112
3.2.3.11	Rancangan Antarmuka Mengisi <i>Feedback</i>	113
3.2.3.12	Rancangan Antarmuka Mengisi <i>Wallet</i>	114
3.2.3.13	Rancangan Antarmuka Melihat Riwayat Konfirmasi <i>Topup Wallet User</i>	115
3.2.3.14	Rancangan Antarmuka Melihat Riwayat Pesan <i>Contact Us</i>	116
3.2.3.15	Rancangan Antarmuka Melihat Riwayat Pemakaian <i>Point</i>	116
3.2.3.16	Rancangan Antarmuka Melihat Riwayat Pemakaian <i>Wallet</i>	117
3.2.3.17	Rancangan Antarmuka Mengelola Data <i>Review Tenant</i>	117

3.2.3.18	Rancangan Antarmuka Mengelola Data <i>Review Event</i>	118
3.2.3.19	Rancangan Antarmuka Menggunakan Fitur <i>Contact Us</i>	119
3.2.3.20	Rancangan Antarmuka Menggunakan Fitur <i>Frequently Asked Questions</i>	119
3.2.3.21	Rancangan Antarmuka <i>Dashboard Admin</i>	120
3.2.3.22	Rancangan Antarmuka Mengisi <i>Review</i> untuk <i>Tenant</i>	121
3.2.3.23	Rancangan Antarmuka Mengisi <i>Review</i> untuk <i>Event</i>	121
3.2.3.24	Rancangan Antarmuka Mendaftar sebagai <i>Event</i> atau <i>Tenant</i>	122
BAB IV IMPLEMENTASI		123
4.1	Lingkungan Implementasi	123
4.1.1	Lingkungan Implementasi Perangkat Keras	123
4.1.2	Lingkungan Implementasi Perangkat Lunak	123
4.2	Implementasi Tampilan Antarmuka	123
4.2.1	Implementasi Mengelola Data <i>Frequently Asked Questions</i>	124
4.2.2	Implementasi Mengelola Pelaporan Keuangan	126
4.2.3	Implementasi Membalas Pesan <i>Contact Us</i>	134
4.2.4	Implementasi Mengelola Data <i>Reward</i> dan <i>Point</i>	135
4.2.5	Implementasi Mengelola Pelaporan Data <i>Tenant</i>	138
4.2.6	Implementasi Mengelola Pelaporan Data <i>Event</i>	140
4.2.7	Implementasi Mengelola Pelaporan Data <i>User</i>	142
4.2.8	Implementasi Melihat Hasil <i>Feedback</i>	143
4.2.9	Implementasi Menggunakan <i>Reward Event</i>	144
4.2.10	Implementasi Menggunakan <i>Reward Tenant</i>	145
4.2.11	Implementasi Mengisi <i>Feedback</i>	146
4.2.12	Implementasi Mengisi <i>Wallet</i>	148
4.2.13	Implementasi Melihat Riwayat Konfirmasi <i>Top-up Wallet</i>	149
4.2.14	Implementasi Melihat Riwayat Pesan <i>Contact Us</i>	150
4.2.15	Implementasi Melihat Riwayat Pemakaian <i>Point</i>	150
4.2.16	Implementasi Melihat Riwayat Pemakaian <i>Wallet</i>	151
4.2.17	Implementasi Mengelola Data <i>Review Tenant</i>	151
4.2.18	Implementasi Mengelola Data <i>Review Event</i>	153
4.2.19	Implementasi Menggunakan Fitur <i>Contact Us</i>	155

4.2.20	Implementasi Menggunakan Fitur <i>Frequently Asked Questions</i>	156
4.2.21	Implementasi Dashboard Admin	156
4.2.22	Implementasi Mengisi <i>Review</i> untuk <i>Tenant</i>	157
4.2.23	Implementasi Mengisi <i>Review</i> untuk <i>Event</i>	157
4.2.24	Implementasi Mendaftar sebagai <i>Event/Tenant</i>	158
4.3	Implementasi Perangkat Lunak	159
4.3.1	Implementasi Proses Admin Mengelola Data <i>Frequently Asked Questions</i>	159
4.3.1.1	Proses Menambah Data <i>Frequently Asked Questions</i>	159
4.3.1.2	Proses Mengubah Data <i>Frequently Asked Questions</i>	160
4.3.1.3	Proses Menghapus Data <i>Frequently Asked Questions</i>	160
4.3.2	Implementasi Proses Admin Mengelola Pelaporan Keuangan	160
4.3.2.1	Proses Melihat Laporan Keuangan Keseluruhan..	161
4.3.2.2	Proses Melihat Laporan Keuangan Bulanan.....	162
4.3.2.3	Proses Melihat Laporan Keuangan Harian	163
4.3.2.4	Proses Mengonfirmasi Pembayaran Pengisian <i>Wallet</i>	164
4.3.2.5	Proses Mengonfirmasi Pembayaran Transaksi Pendaftaran <i>Tenant</i> pada <i>Event</i>	165
4.3.3	Implementasi Proses Admin Membalas Pesan <i>Contact Us</i>	165
4.3.4	Implementasi Proses Admin Mengelola Data <i>Reward</i> dan <i>Point</i>	166
4.3.4.1	Proses Menambah Data <i>Reward</i>	166
4.3.4.2	Proses Mengubah Data <i>Reward</i>	167
4.3.4.3	Proses Menghapus Data <i>Reward</i>	167
4.3.4.4	Proses Melihat <i>Reward</i> yang Dipakai <i>Event</i>	167
4.3.4.5	Proses Melihat <i>Reward</i> yang Dipakai <i>Tenant</i>	168
4.3.4.6	Proses Melihat Data <i>Point</i>	169
4.3.5	Implementasi Proses Admin Mengelola Pelaporan Data <i>Tenant</i>	169
4.3.5.1	Proses Melihat Data <i>Tenant</i>	169
4.3.5.2	Proses Melihat Data <i>Review Tenant</i>	170

4.3.6	Implementasi Proses Admin Mengelola Pelaporan Data <i>Event</i>	171
4.3.6.1	Proses Melihat Data <i>Event</i>	171
4.3.6.2	Proses Melihat Data <i>Review Event</i>	171
4.3.7	Implementasi Proses Admin Mengelola Pelaporan Data User	173
4.3.8	Implementasi Proses Admin Melihat Hasil <i>Feedback</i> ..	173
4.3.9	Implementasi Proses <i>Event</i> Menggunakan <i>Reward Event</i>	174
4.3.10	Implementasi Proses <i>Tenant</i> Menggunakan <i>Reward</i> <i>Tenant</i>	175
4.3.11	Implementasi Proses <i>Event</i> Mendapatkan <i>Point</i>	175
4.3.12	Implementasi Proses <i>Tenant</i> Mendapatkan <i>Point</i>	176
4.3.13	Implementasi Proses User Mengisi <i>Feedback</i>	177
4.3.14	Implementasi <i>Pop-up</i> untuk User Mengisi <i>Feedback</i> ...	178
4.3.15	Implementasi Proses <i>Tenant</i> Mengisi <i>Wallet</i>	179
4.3.16	Implementasi Proses <i>Tenant</i> Melihat Riwayat Konfirmasi <i>Topup Wallet</i>	180
4.3.17	Implementasi Proses User Melihat Riwayat Pesan <i>Contact</i> <i>Us</i>	180
4.3.18	Implementasi Proses <i>User</i> Melihat Riwayat Pemakaian <i>Point</i>	181
4.3.19	Implementasi Proses <i>User</i> Melihat Riwayat Pemakaian <i>Wallet</i>	181
4.3.20	Implementasi Proses <i>Tenant</i> Mengelola Data <i>Review</i> <i>Tenant</i>	182
4.3.21	Implementasi Proses <i>Event</i> Mengelola Data <i>Review Event</i>	182
4.3.22	Implementasi Proses Pengguna Umum Menggunakan Fitur <i>Contact Us</i>	183
4.3.23	Implementasi Proses Pengguna Umum Menggunakan Fitur <i>Frequently Asked Questions</i>	184
4.3.24	Implementasi Proses <i>Event</i> Mengisi <i>Review</i> untuk <i>Tenant</i>	184
4.3.25	Implementasi Proses <i>Tenant</i> Mengisi <i>Review</i> untuk <i>Event</i>	185
4.3.26	Implementasi <i>Pop-up</i> Pengguna Umum untuk Mendaftar sebagai <i>Event</i> atau <i>Tenant</i>	186
BAB V UJI COBA DAN EVALUASI		187

5.1	Lingkungan Pengujian	187
5.2	Skenario Pengujian	187
5.2.1	Pengujian Fungsionalitas	187
5.2.2	Pengujian Ketertarikan Responden terhadap CRM	188
5.2.2.1	Demografi Responden	188
5.2.2.2	Hasil Kuesioner	190
5.3	Evaluasi Pengujian.....	194
5.3.1	Evaluasi Pengujian Fungsionalitas	195
5.3.2	Evaluasi Pengujian Ketertarikan Responden terhadap CRM	196
BAB VI KESIMPULAN DAN SARAN.....		201
5.4	Kesimpulan	201
5.5	Saran	201
DAFTAR PUSTAKA		203
LAMPIRAN.....		207
BIODATA PENULIS.....		219

(Halaman ini sengaja dikosongkan)

RBTC

DAFTAR GAMBAR

Gambar 3.1 Alur Transaksi Utama dalam CallTenant	25
Gambar 3.2 Alur Pengisian Wallet	26
Gambar 3.3 Alur Penukaran Reward	26
Gambar 3.4 Use Case pada User (Event dan Tenant)	31
Gambar 3.5 Use Case pada Pengguna Umum.....	32
Gambar 3.6 Use Case pada Admin	32
Gambar 3.7 Diagram Aktivitas Menambah Data Frequently Asked Questions.....	35
Gambar 3.8 Diagram Aktivitas Mengubah Data Frequently Asked Questions.....	36
Gambar 3.9 Diagram Aktivitas Menghapus Data Frequently Asked Questions.....	38
Gambar 3.10 Diagram Aktivitas Melihat Laporan Keuangan dari Transaksi Pendaftaran	40
Gambar 3.11 Diagram Aktivitas Melihat Laporan Keuangan dari Pengisian Wallet User	42
Gambar 3.12 Diagram Aktivitas Mengonfirmasi Pembayaran Transaksi Pendaftaran	44
Gambar 3.13 Diagram Aktivitas Mengonfirmasi Pembayaran Pengisian Wallet.....	46
Gambar 3.14 Diagram Aktivitas Membalas Pesan Contact Us	48
Gambar 3.15 Diagram Aktivitas Menambah Data Reward.....	50
Gambar 3.16 Diagram Aktivitas Mengubah Data Reward	52
Gambar 3.17 Diagram Aktivitas Menghapus Data Reward	54
Gambar 3.18 Diagram Aktivitas Melihat Data Reward yang Dipakai Event	55
Gambar 3.19 Diagram Aktivitas Melihat Data Reward yang Dipakai Tenant	56
Gambar 3.20 Diagram Aktivitas Melihat Data Point User.....	57
Gambar 3.21 Diagram Aktivitas Melihat Data Tenant	58
Gambar 3.22 Diagram Aktivitas Melihat Data Review Tenant	60
Gambar 3.23 Diagram Aktivitas Melihat Data Event	61
Gambar 3.24 Diagram Aktivitas Melihat Data Review Event	63
Gambar 3.25 Diagram Aktivitas Melihat Data User	64
Gambar 3.26 Diagram Aktivitas Melihat Hasil Feedback	66
Gambar 3.27 Diagram Aktivitas Menggunakan Reward Event	68
Gambar 3.28 Diagram Aktivitas Menggunakan Reward Tenant	70

Gambar 3.29 Diagram Aktivitas Event Mendapatkan Point.....	72
Gambar 3.30 Diagram Aktivitas Tenant Mendapatkan Point.....	74
Gambar 3.31 Diagram Aktivitas Mengisi Feedback.....	76
Gambar 3.32 Diagram Aktivitas Mengisi Wallet	78
Gambar 3.33 Diagram Aktivitas Melihat Riwayat Konfirmasi Topup Wallet	80
Gambar 3.34 Diagram Aktivitas Melihat Riwayat Pesan Contact Us ...	81
Gambar 3.35 Diagram Aktivitas Melihat Riwayat Pemakaian Point.....	82
Gambar 3.36 Diagram Aktivitas Melihat Riwayat Pemakaian Point.....	83
Gambar 3.37 Diagram Aktivitas Mengelola Data Review Tenant	85
Gambar 3.38 Diagram Aktivitas Mengelola Data Review Event	87
Gambar 3.39 Diagram Aktivitas Menggunakan Fitur Contact Us.....	89
Gambar 3.40 Diagram Aktivitas Menggunakan Fitur Frequently Asked Questions	90
Gambar 3.41 Diagram Aktivitas Mengisi Review untuk Tenant.....	92
Gambar 3.42 Diagram Aktivitas Mengisi Review untuk Event.....	94
Gambar 3.43 Diagram Aktivitas Mendaftar sebagai Event/Tenant	96
Gambar 3.44 Rancangan Antarmuka Halaman Frequently Asked Questions	97
Gambar 3.45 Rancangan Antarmuka Menambah Data Frequently Asked Questions	98
Gambar 3.46 Rancangan Antarmuka Mengubah Data Frequently Asked Questions	98
Gambar 3.47 Rancangan Antarmuka Laporan Keseluruhan Laporan Keuangan	99
Gambar 3.48 Rancangan Antarmuka Data Transaksi User	99
Gambar 3.49 Rancangan Antarmuka Detail Data Transaksi User.....	100
Gambar 3.50 Rancangan Antarmuka Halaman Data Wallet User.....	100
Gambar 3.51 Rancangan Antarmuka Detail Data Wallet User.....	101
Gambar 3.52 Rancangan Antarmuka Konfirmasi Pembayaran Transaksi User.....	101
Gambar 3.53 Rancangan Antarmuka Konfirmasi Topup Wallet.....	102
Gambar 3.54 Rancangan Antarmuka Detail Konfirmasi Pembayaran Transaksi User dan Konfirmasi Pembayaran Topup Wallet.....	102
Gambar 3.55 Rancangan Antarmuka Halaman Data Contact Us	103
Gambar 3.56 Rancangan Antarmuka Formulir Balas Pesan Contact Us	103
Gambar 3.57 Rancangan Antarmuka Data Point User.....	104
Gambar 3.58 Rancangan Antarmuka Data Reward	104

Gambar 3.59 Rancangan Antarmuka Menambah Data Reward.....	105
Gambar 3.60 Rancangan Antarmuka Mengubah Data Reward	105
Gambar 3.61 Rancangan Antarmuka Data Reward yang Dipakai Event	106
Gambar 3.62 Rancangan Antarmuka Data Reward yang Dipakai Tenant	106
Gambar 3.63 Rancangan Antarmuka Data Tenant.....	107
Gambar 3.64 Rancangan Antarmuka Data Review Tenant.....	107
Gambar 3.65 Rancangan Antarmuka Detail Data Review Tenant	108
Gambar 3.66 Rancangan Antarmuka Data Event	108
Gambar 3.67 Rancangan Antarmuka Data Review Event	109
Gambar 3.68 Rancangan Antarmuka Detail Data Review Event.....	109
Gambar 3.69 Rancangan Antarmuka Data User	110
Gambar 3.70 Rancangan Antarmuka Data Feedback.....	110
Gambar 3.71 Rancangan Antarmuka Detail Hasil Feedback	111
Gambar 3.72 Rancangan Antarmuka Halaman Reward Event	111
Gambar 3.73 Rancangan Antarmuka Formulir Menggunakan Reward Event	112
Gambar 3.74 Rancangan Antarmuka Formulir Menggunakan Reward Tenant	113
Gambar 3.75 Rancangan Antarmuka Formulir Mengisi Feedback ...	113
Gambar 3.76 Rancangan Antarmuka Formulir Pengisian Wallet	114
Gambar 3.77 Rancangan Antarmuka Halaman Konfirmasi Pengisian Wallet.....	114
Gambar 3.78 Rancangan Antarmuka Formulir Konfirmasi Pengisian Wallet.....	115
Gambar 3.79 Rancangan Antarmuka Halaman Riwayat Konfirmasi Topup Wallet	115
Gambar 3.80 Rancangan Antarmuka Halaman Riwayat Pesan Contact Us	116
Gambar 3.81 Rancangan Antarmuka Halaman Riwayat Pemakaian Point	116
Gambar 3.82 Rancangan Antarmuka Riwayat Pemakaian Wallet	117
Gambar 3.83 Rancangan Antarmuka Halaman Data Review Tenant .	118
Gambar 3.84 Rancangan Antarmuka Halaman Data Review Event ...	118
Gambar 3.85 Rancangan Antarmuka Menggunakan Fitur Contact Us	119
Gambar 3.86 Rancangan Antarmuka Menggunakan Fitur Frequently Asked Questions	119
Gambar 3.87 Rancangan Antarmuka Dashboard Admin	120

Gambar 3.88 Rancangan Antarmuka Mengisi Review untuk Tenant..	121
Gambar 3.89 Rancangan Antarmuka Mengisi Review untuk Tenant..	121
Gambar 3.90 Rancangan Antarmuka Mendaftar Sebagai Event/Tenant	122
Gambar 4.1 Implementasi Halaman Frequently Asked Questions	124
Gambar 4.2 Implementasi Menambah Data Frequently Asked Questions	124
Gambar 4.3 Implementasi Mengubah Data Frequently Asked Questions	125
Gambar 4.4 Implementasi Menghapus Data Frequently Asked Questions	125
Gambar 4.5 Implementasi Laporan Keseluruhan Laporan Keuangan .	126
Gambar 4.6 Implementasi Laporan Bulanan Laporan Keuangan Transaksi Pendaftaran	127
Gambar 4.7 Implementasi Laporan Harian Laporan Keuangan Transaksi Pendaftaran	128
Gambar 4.8 Rancangan Antarmuka Laporan Bulanan Laporan Keuangan Pengisian Wallet	129
Gambar 4.9 Implementasi Laporan Harian Laporan Keuangan Pengisian Wallet	129
Gambar 4.10 Implementasi Halaman Data Transaksi User	130
Gambar 4.11 Implementasi Halaman Detail Data Transaksi User	130
Gambar 4.12 Implementasi Detail Konfirmasi Pembayaran Transaksi User.....	131
Gambar 4.13 Implementasi Data Wallet User	131
Gambar 4.14 Implementasi Konfirmasi Pembayaran Transaksi User .	132
Gambar 4.15 Implementasi Detail Data Wallet User	132
Gambar 4.16 Implementasi Mengonfirmasi Topup Wallet	133
Gambar 4.17 Implementasi Detail Konfirmasi Pembayaran Topup Wallet	133
Gambar 4.18 Rancangan Antarmuka Halaman Data Contact Us	134
Gambar 4.19 Implementasi Membalas Pesan Contact Us	134
Gambar 4.20 Implementasi Data Point User	135
Gambar 4.21 Implementasi Detail Data Point User.....	135
Gambar 4.22 Implementasi Data Reward	136
Gambar 4.23 Implementasi Menambah Data Reward	137
Gambar 4.24 Implementasi Mengubah Data Reward	137
Gambar 4.25 Implementasi Data Reward yang Dipakai Tenant.....	138
Gambar 4.26 Implementasi Data Reward yang Dipakai Event	138

Gambar 4.27 Implementasi Data Tenant.....	139
Gambar 4.28 Implementasi Data Review Tenant	140
Gambar 4.29 Implementasi Detail Data Review Tenant.....	140
Gambar 4.30 Implementasi Data Event	141
Gambar 4.31 Implementasi Data Review Event	141
Gambar 4.32 Implementasi Detail Data Review Event.....	142
Gambar 4.33 Implementasi Data User	142
Gambar 4.34 Implementasi Hasil Feedback	143
Gambar 4.35 Implementasi Detail Hasil Feedback.....	143
Gambar 4.36 Implementasi Halaman Reward Event	144
Gambar 4.37 Implementasi Formulir Menggunakan Reward Event...	144
Gambar 4.38 Implementasi Halaman Reward Tenant	145
Gambar 4.39 Implementasi Formulir Menggunakan Reward Tenant .	145
Gambar 4.40 Implementasi Popup Mengisi Feedback.....	146
Gambar 4.41 Implementasi Formulir Mengisi Feeedback	147
Gambar 4.42 Implementasi Formulir Pengisian Wallet	148
Gambar 4.43 Implementasi Halaman Konfirmasi Pengisian Wallet...	148
Gambar 4.44 Implementasi Formulir Konfirmasi Pengisian Wallet...	149
Gambar 4.45 Implementasi Halaman Riwayat Konfirmasi Topup Wallet	149
Gambar 4.46 Implementasi Halaman Riwayat Pesan Contact Us.....	150
Gambar 4.47 Implementasi Halaman Riwayat Pemakaian Point.....	150
Gambar 4.48 Implementasi Halaman Riwayat Pemakaian Wallet.....	151
Gambar 4.49 Implementasi Profil Tenant	151
Gambar 4.50 Implementasi Halaman Data Review Tenant	152
Gambar 4.51 Implementasi Formulir Balas Review dari Event pada Tenant	152
Gambar 4.52 Implementasi Data Balasan Review dari Event pada Tenant	153
Gambar 4.53 Implementasi Profil Event.....	153
Gambar 4.54 Implementasi Halaman Data Review Event	154
Gambar 4.55 Implementasi Halaman Formulir Balas Review dari Event pada Tenant.....	154
Gambar 4.56 Implementasi Data Balasan Review dari Event pada Tenant	155
Gambar 4.57 Implementasi Halaman Contact Us	155
Gambar 4.58 Implementasi Halaman Frequently Asked Questions....	156
Gambar 4.59 Implementasi Dashboard Admin	156
Gambar 4.60 Implementasi Mengisi Review untuk Tenant	157

Gambar 4.61 Implementasi Mengisi Review untuk Event158

Gambar 4.62 Implementasi Popup Pendaftaran.....158

Gambar 4.63 Implementasi Mendaftar sebagai Event/Tenant159

RBTC

DAFTAR TABEL

Tabel 2.1 Perbandingan Modul CRM CallTenant dengan CRM Sejenis	9
Tabel 3.1 Modul CRM pada Aplikasi	20
Tabel 3.2 Kebutuhan Fungsional	27
Tabel 3.3 Kasus Penggunaan	33
Tabel 3.4 Menambah Data Frequently Asked Questions	34
Tabel 3.5 Mengubah Data Frequently Asked Questions	35
Tabel 3.6 Menghapus Data Frequently Asked Questions	37
Tabel 3.7 Melihat Laporan Keuangan dari Transaksi Pendaftaran	39
Tabel 3.8 Melihat Laporan Keuangan dari Pengisian Wallet User	41
Tabel 3.9 Mengonfirmasi Pembayaran Transaksi Pendaftaran	43
Tabel 3.10 Mengonfirmasi Pembayaran Pengisian Wallet	45
Tabel 3.11 Membalas Pesan Contact Us	47
Tabel 3.12 Menambah Data Reward	49
Tabel 3.13 Mengubah Data Reward	51
Tabel 3.14 Menghapus Data Reward	53
Tabel 3.15 Melihat Data Reward yang Dipakai Event	55
Tabel 3.16 Melihat Data Reward yang Dipakai Tenant	55
Tabel 3.17 Melihat Data Point User	56
Tabel 3.18 Melihat Data Tenant	58
Tabel 3.19 Melihat Data Review Tenant	59
Tabel 3.20 Melihat Data Event	61
Tabel 3.21 Melihat Data Review Event	62
Tabel 3.22 Melihat Data User	64
Tabel 3.23 Melihat Hasil Feedback	65
Tabel 3.24 Menggunakan Reward Event	67
Tabel 3.25 Menggunakan Reward Tenant	69
Tabel 3.26 Event Mendapatkan Point	71
Tabel 3.27 Tenant Mendapatkan Point	73
Tabel 3.28 Mengisi Feedback	75
Tabel 3.29 Mengisi Wallet	77
Tabel 3.30 Melihat Riwayat Konfirmasi Topup Wallet	79
Tabel 3.31 Melihat Riwayat Pesan Contact Us	80
Tabel 3.32 Melihat Riwayat Pemakaian Point	81
Tabel 3.33 Melihat Riwayat Pemakaian Wallet	82
Tabel 3.34 Mengelola Data Review Tenant	84
Tabel 3.35 Mengelola Data Review Event	86
Tabel 3.36 Menggunakan Fitur Contact Us	88

Tabel 3.37 Menggunakan Fitur Frequently Asked Questions	89
Tabel 3.38 Mengisi Review untuk Tenant	91
Tabel 3.39 Mengisi Review untuk Event.....	93
Tabel 3.40 Mendaftar sebagai Event/Tenant	95
Tabel 5.1 Distribusi Responden Berdasarkan Umur.....	189
Tabel 5.2 Distribusi Responden Berdasarkan Pekerjaan	189
Tabel 5.3 Distribusi Responden Berdasarkan Pengalaman Menjadi Event/Tenant.....	189
Tabel 5.4 Distribusi Responden Berdasarkan Pendapatan Rata-Rata per Bulan.....	190
Tabel 5.5 Distribusi Responden Berdasarkan Jumlah Event yang Pernah Diikuti.....	190
Tabel 5.6 Hasil Kuesioner Responden Tenant.....	191
Tabel 5.7 Hasil Kuesioner Responden Event	192
Tabel 5.8 Hasil Kuesioner Responden Admin	193
Tabel 5.9 Evaluasi Pengujian Fungsionalitas	195
Tabel 5.10 Tabel Nilai Z.....	196
Tabel 5.11 Hasil Evaluasi Kuesioner Responden Tenant	198
Tabel 5.12 Hasil Evaluasi Kuesioner Responden Event.....	199
Tabel 5.13 Hasil Evaluasi Kuesioner Responden Admin	200

DAFTAR KODE PROGRAM

Kode Sumber 4.1 Fungsi Menambah Data Frequently Asked Questions	160
Kode Sumber 4.2 Fungsi Mengubah Data Frequently Asked Questions	160
Kode Sumber 4.3 Fungsi Menghapus Data Frequently Asked Questions	160
Kode Sumber 4.4 Fungsi Melihat Laporan Keuangan Keseluruhan ...	161
Kode Sumber 4.5 Fungsi Melihat Laporan Keuangan Bulanan dari Transaksi Pendaftaran	162
Kode Sumber 4.6 Fungsi Melihat Laporan Keuangan Bulanan dari Pengisian Wallet	163
Kode Sumber 4.7 Fungsi Melihat Laporan Keuangan Harian dari Transaksi Pendaftaran	163
Kode Sumber 4.8 Fungsi Melihat Laporan Keuangan Harian dari Pengisian Wallet	164
Kode Sumber 4.9 Fungsi Mengonfirmasi Pengisian Wallet	164
Kode Sumber 4.10 Fungsi Mengonfirmasi Pendaftaran Tenant pada Event	165
Kode Sumber 4.11 Fungsi Membalas Pesan Contact Us	166
Kode Sumber 4.12 Fungsi Menambah Data Reward	166
Kode Sumber 4.13 Fungsi Mengubah Data Reward	167
Kode Sumber 4.14 Fungsi Menghapus Data Reward	167
Kode Sumber 4.15 Fungsi Melihat Reward yang Dipakai Event	168
Kode Sumber 4.16 Fungsi Melihat Reward yang Dipakai Tenant	168
Kode Sumber 4.17 Fungsi Melihat Data Point	169
Kode Sumber 4.18 Fungsi Melihat Data Tenant	170
Kode Sumber 4.19 Fungsi Melihat Review untuk Tenant	170
Kode Sumber 4.20 Fungsi Melihat Detail <i>Review</i> untuk <i>Tenant</i>	171
Kode Sumber 4.21 Fungsi Melihat Data Event	171
Kode Sumber 4.22 Fungsi Melihat Review untuk Event	172
Kode Sumber 4.23 Fungsi Melihat Detail Review untuk Event	172
Kode Sumber 4.24 Fungsi Melihat Data User	173
Kode Sumber 4.25 Fungsi Melihat Hasil Feedback	173
Kode Sumber 4.26 Fungsi Event Menggunakan Reward	174
Kode Sumber 4.27 Fungsi Tenant Menggunakan Reward	175
Kode Sumber 4.28 Fungsi Event Mendapatkan Point	176
Kode Sumber 4.29 Fungsi Tenant Mendapatkan Point	177

Kode Sumber 4.30 Fungsi User Mengisi Feedback.....	178
Kode Sumber 4.31 Implementasi Pop-up untuk User Mengisi Feedback	178
Kode Sumber 4.32 Fungsi Menambah Data Wallet	179
Kode Sumber 4.33 Fungsi Mengisi Konfirmasi Pembayaran Pengisian Wallet	180
Kode Sumber 4.34 Fungsi Tenant Melihat Riwayat Konfirmasi Topup Wallet	180
Kode Sumber 4.35 Fungsi User Melihat Riwayat Pesan Contact Us ..	180
Kode Sumber 4.36 Fungsi User Melihat Riwayat Pemakaian Point....	181
Kode Sumber 4.37 Fungsi User Melihat Riwayat Pemakaian Wallet .	182
Kode Sumber 4.38 Fungsi Tenant Mengelola Data Review Tenant....	182
Kode Sumber 4.39 Fungsi Event Mengelola Data Review Event	183
Kode Sumber 4.40 Fungsi Pengguna Umum Menggunakan Fitur Contact Us.....	183
Kode Sumber 4.41 Fungsi Pengguna Umum Menggunakan Fitur Frequently Asked Questions.....	184
Kode Sumber 4.42 Implementasi Proses Event Mengisi Review untuk Tenant.....	185
Kode Sumber 4.43 Implementasi Proses Tenant Mengisi Review untuk Event.....	186
Kode Sumber 4.44 Implementasi Pop-up Pengguna Umum Mendaftar sebagai Event/Tenant.....	186

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi seperti sekarang ini, berkembangnya teknologi dan mudahnya mengakses pengetahuan memicu kompetisi antar ide dan inovasi manusia menjadi semakin pesat. Terlahir sebagai generasi Z tentu saja merupakan suatu karakteristik dan kelebihan tersendiri. Generasi Z yang disebut juga sebagai iGeneration atau generasi internet sejak kecil sudah mengenal teknologi dan akrab dengan *gadget* canggih yang secara tidak langsung berpengaruh terhadap tingkah lakunya. Generasi Z pada dasarnya memiliki perbedaan dengan generasi Y karena mereka cenderung memilih untuk membuka suatu usaha dibandingkan menjadi pegawai di suatu perusahaan. Dengan kata lain, generasi Z merupakan pencetus lahirnya *startup-startup* masa kini [1].

Salah satu bisnis yang sedang diminati kaum muda saat ini adalah bisnis *event organizer* [2]. Biasanya, *Event Organizer* akan mengadakan *bazaar* atau acara yang diselenggarakan untuk beberapa hari. Namun, sampai saat ini masih banyak *entrepreneur* muda yang belum berkesempatan mengikuti *event* tersebut sebagai *tenant*. Selama ini persebaran informasi terbesar hanya melalui sosial media, sehingga timbul masalah kurangnya informasi mengenai *event* tersebut bagi *tenant*, sulitnya mencari *tenant* yang cocok bagi *event*, sulitnya *event* dan *tenant* untuk menjangkau lokasinya masing-masing, dan masih banyak permasalahan lainnya. Oleh karena adanya keterbatasan komunikasi dan publikasi, adanya suatu aplikasi yang dapat mengintegrasikan data-data *event* dan *tenant* tersebut ke dalam suatu wadah akan sangat berguna untuk memperluas publikasi dan komunikasi antar *event* dan *tenant* tersebut, sehingga dapat meningkatkan potensi bisnis yang sudah ada.

CallTenant adalah suatu aplikasi yang bertujuan untuk membantu memudahkan komunikasi antar *event* dan *tenant* dengan layanan dan fitur yang disediakan. Layaknya Go-Jek yang bertujuan untuk membantu memudahkan masyarakat mencari kendaraan umum, aplikasi CallTenant sendiri juga memiliki tujuan sosial membantu mengefisienkan kegiatan masyarakat sehingga CallTenant membutuhkan CRM yang merupakan fungsi terintegrasi yang menunjang sebuah perusahaan dalam bidang pemasaran, penjualan, dan pelayanan sehingga dapat meningkatkan nilai kepuasan pelanggan. Berdasarkan tiga tahap penting, yaitu *Acquire*, *Enhance*, dan *Retain*, *Customer Relationship Management* (CRM) sangat dibutuhkan dalam membangun suatu perusahaan karena berfokus pada merawat membangun, dan mengelola hubungan antara perusahaan atau organisasi dan *stakeholder* [3]. Tujuan dari semua kegiatan CRM adalah untuk memperoleh dan meningkatkan nilai pelanggan (kepuasan pelanggan dan loyalitas pelanggan) serta profitabilitas pelanggan selama seluruh waktu hubungan pelanggan (*customer lifetime value*).

1.2 Rumusan Masalah

1. Bagaimana perancangan *Customer Relationship Management* pada aplikasi CallTenant menarik bagi konsumen?
2. Bagaimana membangun fitur CRM bagi CallTenant sesuai dengan tahapan dan tipe CRM?

1.3 Batasan Permasalahan

1. Desain CRM ditujukan pada aplikasi CallTenant yang belum berjalan secara operasional.
2. Tidak ada pengukuran performa antara aplikasi CallTenant dengan CRM atau tidak dengan CRM karena belum berjalan secara operasional.
3. Survei pengujian diambil dari hasil kuesioner yang diisi oleh masyarakat umum yang berlangsung pada bulan Desember.

4. CRM dibangun berdasarkan aplikasi CallTenant yang berbasis web dengan *framework* Laravel.

1.4 Tujuan

Menghasilkan rancangan dan desain CRM yang dapat digunakan sesuai dengan kebutuhan aplikasi CallTenant.

1.5 Manfaat

1. Membantu dan meningkatkan kualitas *marketing* perusahaan dalam mengidentifikasi pelanggan secara tepat.
2. Membantu perusahaan dalam meningkatkan kinerja pegawai dengan mempersingkat proses kerja yang ada sebelumnya.
3. Memungkinkan bentuk hubungan personal dengan pelanggan untuk meningkatkan kepuasan pelanggan dan memaksimalkan keuntungan yang didapat.
4. Mengidentifikasi pelanggan yang potensial dan memberikan mereka servis yang lebih dibandingkan pelanggan lainnya.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahap pertama dalam proses pengerjaan tugas akhir ini adalah menyusun proposal tugas akhir. Pada proposal tugas akhir ini diajukan beberapa metode dan *tools* yang cocok untuk menyelesaikan masalah dari tugas akhir ini.

1.6.2 Studi Literatur

Literatur mengenai CRM baik dalam aspek informatika maupun di luar informatika diperlukan untuk menyusun CRM yang baik. Materi dapat diambil dari contoh tugas akhir, internet, buku, maupun jurnal. Selain itu, akan diadakan survei yang diambil dari hasil kuesioner yang akan diisikan oleh masyarakat umum.

1.6.3 Analisis dan Desain Perangkat Lunak

Dimulai dengan spesifikasi kebutuhan CallTenant sehingga dapat ditemukan kelebihan dan kelemahannya juga. Kemudian dari kebutuhan-kebutuhan yang diperlukan oleh CallTenant, dapat dihasilkan aspek-aspek penting yang mempengaruhi kepuasan seorang pelanggan. Aspek-aspek tersebut yang nantinya akan diolah ke dalam uji coba aplikasi CRM.

1.6.4 Implementasi Perangkat Lunak

Pembuatan CRM pada aplikasi CallTenant menggunakan bahasa pemrograman PHP, HTML, CSS, JavaScript [4]. Pembuatan CRM ini juga menggunakan *tools* Sublime Text, *Framework* Laravel, basis data MySQL, dan XAMPP [5].

1.6.5 Pengujian dan Evaluasi

Pengujian akan dilakukan dengan cara diujicobakan secara mandiri terhadap fungsionalitasnya, kemudian kepada beberapa responden yang terjun di bidang *event organizer* dan yang pernah memiliki/memiliki usaha atau toko. Hasil uji coba dari responden tersebut diolah dalam kuesioner [6] terkait dengan ketertarikan CRM terhadap responden.

1.6.6 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
2. Bab II. Tinjauan Pustaka
3. Bab III. Perancangan Perangkat Lunak
4. Bab IV. Implementasi

5. Bab V. Hasil Uji Coba dan Evaluasi
6. Bab VI. Kesimpulan dan Saran
7. Daftar Pustaka
8. Lampiran

RBTC

(Halaman ini sengaja dikosongkan)

RBTC

BAB II

DASAR TEORI

Bab ini berisi pembahasan mengenai teori-teori dasar atau penjelasan dari metode, algoritma, *library* dan *tools* yang digunakan dalam Tugas Akhir.

2.1 Aplikasi Sejenis

Sebelum merancang dan mengimplementasikan sistem, peninjauan aplikasi serupa dilakukan guna menentukan standardisasi dalam sistem informasi *event* dan *tenant*. Aplikasi yang dipilih adalah aplikasi PusatBazaar.com

2.1.1 Pusatbazaar.com

Pusatbazaar.com adalah *website* yang digunakan sebagai pusat informasi dari *event* dan *tenant*, di mana kedua belah pihak dapat bertukar informasi dalam satu *website* [7]. Pada aplikasi PusatBazaar.com pengguna dapat melakukan pencarian *event* dan *tenant* yang diinginkan. Selain itu, pengguna juga dapat melihat *blog-blog* terkait dengan *event* dan *tenant* tersebut.

Pusatbazaar.com menyediakan banyak fitur lain di antaranya pengguna dapat melihat lowongan pekerjaan dan melihat artikel terkait. Tetapi untuk proses pendaftaran masih dilakukan secara manual yaitu dengan cara menghubungi langsung pemilik *tenant* atau penyelenggara *event* tersebut. Aplikasi Pusatbazaar.com tidak menyediakan fitur transaksi *online* bagi pengguna di mana tidak terdapat juga fitur *wallet* atau *point* yang digunakan untuk pembayaran. Bagi pengguna *event*, tidak terdapat formulir pendaftaran yang biasanya dibuat sebagai persyaratan seleksi dari *tenant*. Aplikasi ini memiliki fitur lain seperti *review* yang dapat diisi oleh pengguna umum sehingga dapat dikatakan *review* pada aplikasi ini kurang terpercaya, karena semua orang yang tidak berhubungan dengan *event* atau *tenant* dapat memberi *review*.

2.2 CRM Sejenis

Sebelum merancang dan mengimplementasikan sistem, meninjau aplikasi serupa dilakukan guna menentukan standarisasi dalam aplikasi CallTenant. Aplikasi yang dipilih adalah ZohoCRM.

2.2.1 ZohoCRM

Zoho adalah *web-based* CRM yang didesain untuk menarik, mempertahankan, dan memuaskan pelanggan untuk meningkatkan bisnis. Fungsionalitas utama ZohoCRM adalah menggunakan manajemen *lead* dan *contact*, manajemen *sales pipeline*, dan *purchase control*. Zoho CRM menyediakan 10 komponen yang berbeda dengan modul utama *navigation panel* yang berguna untuk memudahkan fungsi *searching*. Namun, pengguna juga dapat mengkustomisasi *dashboard* menjadi lebih detail sesuai kebutuhan bisnis.

ZohoCRM dapat dipakai oleh bisnis kecil, pengusaha, maupun perusahaan besar. ZohoCRM dapat mengotomatiskan aktivitas bisnis harian, *track* penjualan, dan menghubungkan dengan pelanggan yang berbeda *platform*. Selain itu, fungsi dari CRM Zoho dapat diperluas dengan mengintegrasikan bersama *Microsoft Outlook*, *MailChimp*, *QuickBooks*, *ConstantConnect*, dan *Google Apps* yang memiliki penghubung *cloud-based* seperti *OneSaas* dan *CloudWork* [8].

2.2.2 Perbandingan CRM CallTenant dengan CRM Sejenis

Beberapa contoh fitur ZohoCRM yang diambil sebagai elemen pembanding dapat dilihat pada Tabel 2.1. Dari sekian banyak fitur yang disediakan oleh ZohoCRM, sebagian besar digunakan juga di dalam CallTenant. Namun, dibangunnya CRM CallTenant dikarenakan tidak semua fitur ZohoCRM dapat diintegrasikan dengan sistem CallTenant atau dengan kata lain tidak fleksibel, seperti adanya fitur *wallet* untuk menyediakan dompet virtual, *reward* tertentu bagi *event/tenant*, *point* dari aktivitas tertentu, dan *review*.

Tabel 2.1 Perbandingan Modul CRM CallTenant dengan CRM Sejenis

	CallTenant	Zoho
Database Backend	MySQL	<i>Proprietary</i>
Bahasa Implementasi	PHP, JavaScript	Java
Web-based	Ya	Ya
Sales Force Automation	Tidak Ada	<i>Leads</i>
	<i>User</i>	<i>Accounts</i>
CRM Analytics	Admin	<i>Standard Dashboard</i>
	Admin	<i>Standard Reports</i>
Manajemen Inventori	<i>Tenant</i>	Produk
	Admin (<i>Financial</i>)	<i>Invoice</i>
	<i>Event</i>	<i>Vendor</i>
	Admin (<i>Financial</i>)	<i>Purchase Order</i>
Customer Support	<i>Contact Us</i>	<i>Cases</i>
	<i>Feedback</i>	<i>Solutions</i>
	<i>FAQ</i>	<i>Visitor History Module</i>
<i>Review</i>	Ya	<i>Custom</i>
<i>Wallet</i>	Ya	Tidak Ada
<i>Point</i>	Ya	<i>Custom</i>
<i>Reward</i>	Ya	<i>Custom</i>

2.3 Event Organizer

Event Organizer adalah sebuah jasa profesional untuk menyelenggarakan sebuah acara dalam ruangan ataupun di luar ruangan. Jasa dan peranan *Event Organizer* saat ini semakin banyak dibutuhkan seiring dengan pesatnya pertumbuhan bisnis yang membutuhkan pengelolaan sebuah acara. Bisnis *Event Organizer* dapat dikatakan bisnis yang memiliki prospek yang sangat bagus dan menguntungkan. Saat ini sudah banyak pelaku usaha baru yang berminat untuk memulai usaha *Event Organizer*, tetapi menurut statistik dari *Small Business Administration* (SBA), data menunjukkan bahwa 50% usaha baru gagal di tahun pertama, 47.5% usaha baru lainnya gagal dalam kurun 5 tahun, dan usaha yang berhasil hanya 2,5%. Semua itu karena pelaku usaha tidak mampu menyelesaikan permasalahan usahanya [9].

2.4 *Tenant*

Sebuah *event* tidak akan luput dari hadirnya *tenant-tenant* untuk menyukseskan acaranya. Secara finansial, *tenant* merupakan sumber pendapatan sebuah *event* dalam pembayaran terhadap biaya investasi maupun untuk pembiayaan secara operasional. Saat ini kebanyakan pemilik dari *tenant* memilih mengikuti *event-event* yang ada karena untuk membuka toko langsung sendiri membutuhkan modal yang besar. Oleh sebab itu, *tenant* selalu mengikuti *event-event* di mall, gedung perkantoran, dll. untuk mengetahui pangsa pasar.

Tenant sendiri biasanya berpartisipasi dalam *event* dengan berbentuk *booth* atau sebuah *stage* mini yang dilengkapi beragam aksesoris yang digunakan sebagai ajang promo produk, jasa, maupun *branding* perusahaan pada sebuah *event*/pameran. *Booth* tersebut berfungsi sebagai tempat promosi untuk menarik konsumen dan menjadi penghubung kegiatan penjual dan pembeli [10].

2.5 *CRM (Customer Relationship Management)*

CRM membuat perusahaan dapat mengelola hubungan bisnis serta data yang dibutuhkan oleh perusahaan tersebut. Dengan CRM, perusahaan dapat menyimpan data pelanggan, informasi penting pelanggan, petunjuk, serta peluang usaha dalam satu tempat atau *cloud* sehingga dapat diakses dengan mudah di mana pun dan kapan pun.

2.5.1 *Konsep CRM*

CRM (*Customer Relationship Management*) merupakan strategi dan usaha untuk menjalin hubungan dengan pelanggan dan memberikan pelayanan yang memuaskan bagi pelanggan. CRM mendukung suatu perusahaan untuk menyediakan pelayanan kepada pelanggan secara *real time* dengan menjalin hubungan dengan tiap pelanggan yang berharga melalui proses penjualan dan layanan, penggunaan informasi tentang pelanggan, *forecasting* serta analisis tren perilaku pelanggan. Berdasarkan apa yang diketahui dari pelanggan, perusahaan dapat membuat variasi penawaran, pelayanan, program, pesan, dan media [11].

2.5.2 Teknologi CRM

Teknologi CRM paling tidak harus memiliki elemen-elemen sebagai berikut ini [12]:

- Aturan-aturan bisnis
Tergantung dari kompleksitas transaksi, aturan-aturan bisnis harus dibuat untuk memastikan bahwa transaksi dengan pelanggan dilakukan dengan efisien.
- Pergudangan data
Konsolidasi dari informasi tentang pelanggan harus dilakukan dalam sistem terpadu.
- Situs/*Webside*
CRM harus memiliki kemampuan swalayan. Aplikasi berbasis web dapat mendukung ini.
- Pelaporan (*Reporting*)
Teknologi CRM harus mampu menghasilkan laporan yang akurat.
- Meja bantu (*Help Desk*)
Teknologi yang mampu mengintegrasikan informasi pelanggan ke aplikasi *help desk* akan menunjukkan keseriusan perusahaan.

2.5.3 Tipe CRM

Menurut Francis Buttle, beberapa perbedaan pendapat mengenai CRM dapat dijelaskan dengan mempertimbangkan adanya sejumlah tipe-tipe yang berbeda dalam mengidentifikasi CRM yaitu [13]:

- *Strategic CRM*
Berkonsentrasi pada pengembangan budaya bisnis *customer-centric* yang bertujuan untuk memenangkan dan mempertahankan konsumen. Budaya bisnis *customer-centric* dapat dicapai dengan menciptakan dan memberikan nilai yang lebih baik dibandingkan dengan kompetitor. Dalam budaya seperti *customer-centric*, umumnya sumber daya yang ada akan ditempatkan pada posisi yang paling meningkatkan nilai konsumen, seperti menerapkan sistem *reward* untuk mendorong perilaku pegawai yang akan meningkatkan kepuasan pelanggan. Selain itu, informasi

mengenai konsumen juga dikumpulkan, dibagikan, dan diterapkan dalam bisnis.

- Operasional CRM
Mencakup otomatisasi yang terintegrasi dari keseluruhan proses bisnis yang terintegrasi, termasuk didalamnya *customer touch point* dan *customer channels* dan integrasi *front office* dan *back office*. Operasional CRM mengotomatiskan proses bisnis yang berinteraksi secara langsung dengan pelanggannya [3].
- Analitis CRM
Analitis CRM merupakan proses analisis dari data-data yang dihasilkan pada operasional CRM.
- Kolaboratif CRM
Aplikasi pelayanan yang berkolaborasi, seperti *email*, *personalized publishing*, *e-communities*, forum diskusi dan sarana lainnya yang dirancang untuk memfasilitasi interaksi antar pelanggan dan pihak perusahaan.

2.5.4 Tahapan CRM

- Mendapatkan pelanggan baru (*Acquire*)
Pelanggan baru didapatkan dengan memberikan kemudahan pengaksesan informasi, inovasi baru, dan pelayanan yang menarik.
- Meningkatkan hubungan dengan pelanggan (*Enhance*)
Perusahaan berusaha menjalin hubungan dengan pelanggan melalui pemberian pelayanan yang baik terhadap pelanggannya.
- Mempertahankan pelanggan (*Retain*)
Usaha mendapatkan loyalitas pelanggan dengan mendengarkan pelanggan dan berusaha memenuhi keinginan pelanggan [12].

2.5.5 Implementasi CRM pada Saat Ini

Saat ini, perusahaan berkembang mengelola hubungan dan informasi pelanggannya dengan berbagai cara. Beberapa masih menggunakan cara lama dengan catatan atau *rolodex* (kumpulan kartu nama), beberapa menggunakan *handphone*, dan beberapa perusahaan lainnya menggunakan *Excel spreadsheets* atau *Google documents*. Metode tersebut dapat digunakan oleh perusahaan dalam jangka

waktu pendek. Tetapi, jika perusahaan ingin melanjutkan untuk memperluas bisnis maka sudah saatnya mempertimbangkan CRM untuk mengumpulkan data-data berharga perusahaan ke dalam satu tempat atau *cloud* agar dapat diakses di mana pun dan kapan pun tanpa resiko bahwa data akan rusak seiring berjalannya waktu.

Sejak dilakukan pengoptimalan komputer dan segala perangkat maupun *software* pendukungnya, perkembangan CRM menjadi sedemikian cepat. *Software* CRM sekarang ini sudah cukup banyak beredar di pasaran, kebanyakan dibuat oleh perusahaan Amerika dan Eropa yang lebih dahulu mengimplementasikan aplikasi CRM di perusahaan mereka. Salah satunya dengan software ZohoCreator [14].

Aplikasi CRM berguna bagi perusahaan dalam banyak hal. Pertama, dalam proses otomatisasi dari seluruh data yang ingin dipakai perusahaan dalam membangun basis data pelanggan. Dapat dibayangkan betapa sulitnya mengumpulkan data-data pelanggan, mencatat berapa kali mereka menghubungi perusahaan dalam satu bulan, berapa kali mereka menggunakan produk atau layanan perusahaan, dan berbagai data lain jika dilakukan secara manual. Kedua, aplikasi CRM memberikan laporan-laporan dari data yang dikumpulkan sehingga dapat menjadi informasi yang berguna bagi manajemen untuk proses pengambilan keputusan. Aplikasi CRM akan menjadi *Decision Support System*, di mana pihak manajemen tidak lagi direpotkan dengan urusan teknis dalam membuat laporan dan menyusun informasi yang dibutuhkan.

2.6 Customer Lifecycle (Siklus Hidup Pelanggan)

Siklus hidup pelanggan adalah istilah yang digunakan untuk menggambarkan perkembangan langkah yang dilakukan pelanggan saat mempertimbangkan, membeli, menggunakan, dan memelihara loyalitas terhadap produk atau layanan. Siklus hidup pelanggan terdiri dari 5 faktor utama, yaitu *reach*, *acquisition*, *conversion*, *retention*, dan *loyalty*. Proses *reach*, *retention*, dan *loyalty* difokuskan menjadi acuan Tugas Akhir ini.

Reach adalah tahap awal dalam siklus hidup pelanggan, terutama dalam hal pemasaran yang berguna untuk mendapatkan

pelanggan. *Acquisition* adalah tahap mendapatkan informasi-informasi penting pelanggan seperti nomor telepon, *e-mail*, dll. Tahap ini menginisiasi hubungan dengan pelanggan yang bisa dilakukan dengan cara mengirimkan *personalised message*, *email marketing*, atau *sales call*. *Conversion* adalah tahap di mana pelanggan menjadi sumber pendapatan atau perusahaan sudah meyakinkan pelanggan bahwa mereka membutuhkan produk atau jasa yang ditawarkan perusahaan tersebut.

Oleh karena itu, perusahaan harus memperhatikan pelanggan untuk mendapatkan kenyamanan dan kepuasan dalam memakai produk atau jasa perusahaan. Di tahap ini juga adalah waktunya untuk menganalisis teknik pemasaran yang sedang dilakukan sehingga dihasilkan teknik pemasaran terbaik dan yang paling cocok dilakukan perusahaan. Mengubah calon pelanggan menjadi pelanggan bukan akhir dari siklus hidup pelanggan atau *customer lifecycle*. Mulai dari tahap *retention*, membangun hubungan pelanggan dengan melibatkannya dalam aktivitas reguler perusahaan yang bertujuan untuk mempertahankan kesetiaan pelanggan yang berharga dengan mengurangi perpindahan pelanggan.

Terakhir adalah tahap *loyalty* yang merupakan pelanggan yang setia merupakan hasil dari pengembangan hubungan pelanggan yang kuat seiring berjalannya tahap siklus hidup pelanggan. Pelanggan yang setia tidak hanya menghasilkan pendapatan bagi perusahaan, namun juga mereka akan mereferensikan perusahaan tersebut kepada kerabat dan temannya [13].

2.7 Customer Retention (Retensi Pelanggan)

Bukanlah hal baru bahwa melibatkan pelanggan merupakan hal krusial yang harus dihadapi untuk melanjutkan penjualan perusahaan dan juga diketahui bahwa jauh lebih mudah dan sedikit membutuhkan biaya dalam berbisnis dengan pelanggan yang sudah ada daripada dengan pelanggan yang baru. Retensi pelanggan adalah pemeliharaan hubungan dengan pelanggan yang dibutuhkan untuk mencegah pelanggan berpindah haluan. Dengan kata lain, retensi pelanggan dimulai dengan hubungan pertama yang dimiliki perusahaan dengan

pelanggan dan berlanjut sepanjang masa hubungan tersebut. Kemampuan perusahaan untuk menarik dan mempertahankan pelanggan baru tidak hanya terkait dengan produk atau layanannya, namun juga dengan cara memberikan layanan kepada pelanggan lama [13].

Retensi pelanggan memiliki dampak langsung terhadap keuntungan. Penelitian oleh John Fleming dan Jim Asplund menyatakan bahwa pelanggan yang banyak terlibat menghasilkan 1,7 kali lebih banyak keuntungan daripada pelanggan biasa, sedangkan memiliki pegawai dan pelanggan yang banyak terlibat mendatangkan keuntungan 3,4 kali lebih besar dari keuntungan normal [15].

2.8 Loyalitas Pelanggan

Pelanggan (*customer*) berbeda dengan konsumen (*consumer*), di mana seseorang dikatakan sebagai pelanggan apabila orang tersebut mulai membiasakan diri untuk membeli produk atau jasa yang ditawarkan oleh suatu badan usaha. Kebiasaan tersebut dapat dibangun melalui pembelian berulang-ulang dalam jangka waktu tertentu. Apabila dalam jangka waktu tertentu tidak melakukan pembelian ulang maka orang tersebut tidak dapat dikatakan sebagai pelanggan tetapi seorang pembeli atau konsumen [16].

Pengertian tentang seorang pelanggan yang loyal menurut Griffin adalah pelanggan yang melakukan pembelian secara berulang-ulang pada badan usaha yang sama secara teratur, membeli produk dan jasa yang ditawarkan oleh badan usaha yang sama, memberitahukan kepada orang lain tentang kepuasan-kepuasan yang didapat dari badan usaha, dan menunjukkan kekebalan terhadap tawaran-tawaran dari badan usaha pesaing [17].

Kotler memberikan definisi yang serupa mengenai loyalitas pelanggan, yaitu suatu pembelian ulang yang dilakukan oleh seorang pelanggan karena komitmen pada suatu merek atau perusahaan. Menurutny ada dua faktor yang mempengaruhi pelanggan untuk loyal, yaitu faktor harga yang membuat pelanggan memilih produk yang murah di antara pilihan yang ada, serta faktor kebiasaan di mana

seseorang yang telah terbiasa menggunakan suatu merek akan sulit untuk berpindah ke merek lain.

Loyalitas pelanggan secara umum dapat diartikan sebagai kesetiaan seseorang atas suatu produk, baik barang maupun jasa tertentu. Loyalitas pelanggan merupakan manifestasi dan kelanjutan dari kepuasan pelanggan dalam menggunakan fasilitas maupun jasa pelayanan yang diberikan oleh perusahaan, serta untuk tetap menjadi pelanggan dari perusahaan tersebut.

Masing-masing pelanggan mempunyai dasar loyalitas yang berbeda, hal ini tergantung dari objektivitas mereka masing-masing. Kotler dan Armstrong mengemukakan bahwa loyalitas berasal dari pemenuhan harapan pelanggan, sedangkan ekspektasi sendiri berasal dari pengalaman pembelian terdahulu oleh pelanggan, opini dari teman dan kerabat, serta janji atau informasi dari pemasar atau pesaing [18].

Pelanggan yang loyal akan dengan mudah membantu menarik pelanggan baru dengan obrolan mulut ke mulut. Hal ini dikarenakan pelanggan loyal sudah tahu persis produk yang digunakan sehingga tidak sulit untuk menceritakannya kepada orang lain. Pelanggan yang loyal juga seiring berjalannya waktu akan membeli lebih banyak, dengan demikian akan semakin sulit untuk berpindah hati kepada produk pesaing. Dengan adanya pelanggan yang loyal, secara tidak langsung akan membantu kelangsungan hidup perusahaan dan juga membantu memasarkan barang atau jasa yang dimiliki perusahaan tersebut [19].

2.9 *Wallet*

Wallet adalah sebuah dompet virtual (*e-wallet*) yang dimiliki setiap pengguna untuk menyimpan dana hasil penjualan (*remit*) dan dana hasil pengembalian (*refund*) transaksi seperti yang sudah ada saat ini pada BukaLapak. Manfaat adanya *Wallet* adalah [20]:

1. Mempercepat proses pengembalian dana (*refund*).
2. Mempercepat transfer dana ke penjual (*remit*).
3. Memberikan alternatif metode pembayaran lebih mudah dan cepat.

Dengan adanya *wallet*, pelanggan tidak harus selalu memiliki uang tunai karena uang tersebut sudah berada dalam data elektronik dan setiap bertransaksi uangnya dipotong sesuai dengan jumlah yang harus dibayarkan sehingga pelanggan tidak perlu khawatir memikirkan uang kembalian. Semua data atau jejak transaksi pelanggan pun juga akan tersimpan dalam basis data sehingga mudah untuk dilacak kembali. Selain itu, dompet virtual ini memiliki beberapa keunggulan dibandingkan dengan layanan perbankan tradisional, yaitu [21]:

1. Penghematan waktu, yaitu pelanggan tidak perlu antri ke ATM atau repot menggunakan m-banking lagi untuk menyelesaikan transaksi karena sudah memiliki uang dalam dompet virtual.
2. Pengendalian biaya berguna bagi pelanggan yang ingin mengendalikan semua pengeluarannya karena dompet virtual ini memiliki sistem rekam jejak sehingga pelanggan dapat memeriksa kapan pun dan di mana pun saat diinginkan.
3. Mengurangi resiko kerugian dan pencurian karena uang tidak berbentuk secara fisik.
4. Mudah digunakan karena sistemnya didesain untuk menjangkau khalayak seluas mungkin.
5. Kenyamanan karena transaksi dapat dilakukan setiap saat dan di mana saja cukup dengan memiliki akses internet.

(Halaman ini sengaja dikosongkan)

RBTC

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada subbab ini akan dijelaskan deskripsi umum aplikasi, deskripsi CRM, analisis proses bisnis, spesifikasi kebutuhan sistem CRM pada CallTenant, serta identifikasi aktor CallTenant.

3.1.1 Deskripsi Umum

CallTenant adalah aplikasi berbasis web yang digunakan sebagai media penghubung bagi penyelenggara acara (*event*) dan pemilik usaha (*tenant*). Aplikasi ini memiliki beberapa macam fitur yang dibuat pada tugas akhir ini ditandai dengan warna hijau pada diagram seperti *Contact Us*, *FAQ*, *Wallet*, *Point*, *Reward*, Laporan, serta *Review*. Diagram selengkapanya dapat dilihat pada Lampiran 4.

3.1.2 Deskripsi Implementasi CRM

Sistem informasi *Customer Relationship Management* dapat digambarkan sebagai sebuah fasilitas yang menyediakan informasi produk seperti *event/tenant*, memberikan informasi yang sedang ditangani dan yang sudah ditangani, menyediakan fasilitas untuk mendekatkan hubungan kerjasama antar pelanggan dan perusahaan.

Selain itu, sistem informasi ini juga mendukung transaksi pendaftaran antar *event* dan *tenant* secara *online* yang dilengkapi fitur konfirmasi pembayaran sehingga *event* maupun *tenant* tidak perlu sulit untuk mencari dan mendatangi lokasi satu sama lain. Sistem ini juga menyediakan fitur *point* dan *reward* bagi pelanggannya sebagai hadiah keloyalitasan.

Dengan adanya perancangan *Customer Relationship Management* berbasis web ini diharapkan akan memudahkan para *tenant* yang ingin mendapatkan informasi tentang *event* begitu juga sebaliknya. Fitur dan modul CRM yang akan dibangun menggunakan tiga tipe CRM, yaitu *strategic*, *analitis*, dan *operasional* atau dengan

kata lain 75% dari tipe CRM yang ada. Penjabaran selengkapnya dijelaskan pada Tabel 3.1.

Tabel 3.1 Modul CRM pada Aplikasi

Modul CRM			Kode Kebutuhan	Kebutuhan Fungsional
Admin (Analitis)	Laporan	FAQ	F-001	Mengelola data <i>Frequently Asked Questions</i>
		<i>Financial</i>	F-002	Mengelola pelaporan keuangan
		<i>Contact Us</i>	F-003	Membalas pesan <i>Contact Us</i>
		<i>Reward dan Point</i>	F-004	Mengelola data <i>reward</i> dan <i>point</i>
		<i>Tenant</i>	F-005	Mengelola pelaporan data <i>tenant</i>
		<i>Event</i>	F-006	Mengelola pelaporan data <i>event</i>
		<i>User</i>	F-007	Mengelola pelaporan data <i>user</i>
		<i>Feedback</i>	F-008	Melihat hasil <i>feedback</i>
Reward (Strategic)	Redeem		F-009	Menggunakan <i>reward event</i>
			F-010	Menggunakan <i>reward tenant</i>

Point (Strategic)		F-011	<i>Event</i> mendapatkan <i>point</i>
		F-012	<i>Tenant</i> mendapatkan <i>point</i>
		F-017	Melihat riwayat pemakaian <i>point</i>
Wallet (Strategic)	<i>Top-up</i>	F-014	Mengisi <i>wallet</i>
	Konfirmasi	F-015	Melihat riwayat konfirmasi <i>to- pup wallet</i>
		F-018	Melihat riwayat pemakaian <i>wallet</i>
Feedback (Analitis)		F-013	Mengisi <i>feedback</i>
Contact Us (Operasional dan Analitis)		F-016	Melihat riwayat pesan <i>Contact Us</i>
		F-021	Menggunakan fitur <i>Contact Uss</i>

FAQ (Operasional)	F-022	Menggunakan fitur <i>Frequently Asked Questions</i>
Review (Operasional dan Strategic)	F-019	Mengelola data <i>review tenant</i>
	F-020	Mengelola data <i>review event</i>
	F-023	Mengisi <i>review</i> untuk <i>tenant</i>
	F-024	Mengisi <i>review</i> untuk <i>event</i>
Registrasi User (Operasional)	F-025	Mendaftar sebagai <i>event/tenant</i>

3.1.3 Tahapan CRM untuk Sistem

Tahapan CRM untuk sistem dibagi menjadi 3 yaitu, *Acquire*, *Enhance*, *Retain*.

3.1.3.1 Tahap Acquire (Memperoleh Pelanggan Baru)

a. Pendaftaran

Fitur ini dirancang untuk menarik pelanggan baru untuk bergabung menjadi anggota untuk mendapatkan fasilitas keanggotaan dengan keuntungan tertentu.

b. Katalog

Bagian katalog akan menyajikan produk *event/tenant* yang dikelompokkan menjadi beberapa kategori misalkan seperti popularitas, ulasan, dll. untuk memudahkan dan mengurangi waktu pengguna dalam mencari informasi produk.

c. *Frequently Asked Questions*

Halaman ini menampilkan tentang pertanyaan yang sering ditanyakan kepada pihak CallTenant sehingga dapat digunakan menjadi informasi awal bagi calon pelanggan.

d. *Contact Us*

Pelanggan yang merasa masih kurang mengerti mengenai sistem atau hal lain yang berhubungan dengan CallTenant dapat menghubungi pihak Calltenant dengan fitur *Contact Us*.

3.1.3.2 Tahap Enhance (Meningkatkan Hubungan dengan Pelanggan)

a. *My Profile*

Pelanggan yang sudah terdaftar menjadi *member* CallTenant dapat mengelola profil yang dimiliki setiap saat demi kebutuhan bisnisnya.

b. *Pembayaran Online*

Event dan *tenant* dapat bertransaksi melalui web CallTenant melalui transfer ATM dengan dilengkapi fitur konfirmasi pembayaran. Segala pembayaran akan masuk ke pihak CallTenant terlebih dahulu sehingga mengurangi terjadinya penipuan, kemudian setelah melalui proses konfirmasi *event* maka *tenant* akan terdaftar dan uang pendaftaran akan masuk ke *wallet event*.

c. *Wallet*

Member CallTenant juga dapat menggunakan fitur *Wallet* sehingga tidak perlu repot melalui transfer ATM untuk melakukan transaksi pembayaran. *Wallet* merupakan dompet virtual yang dapat dipakai dalam web CallTenant.

d. *Review*

Event yang terdaftar dalam CallTenant dapat menggunakan fitur *review* yang ditujukan kepada *tenant* yang mendaftarkan diri padanya dan sebaliknya bagi *tenant* dapat menggunakan fitur *review* yang ditujukan kepada *event* yang pernah didaftarkan olehnya.

3.1.3.3 Tahap Retain (Mempertahankan Pelanggan)

a. *Point*

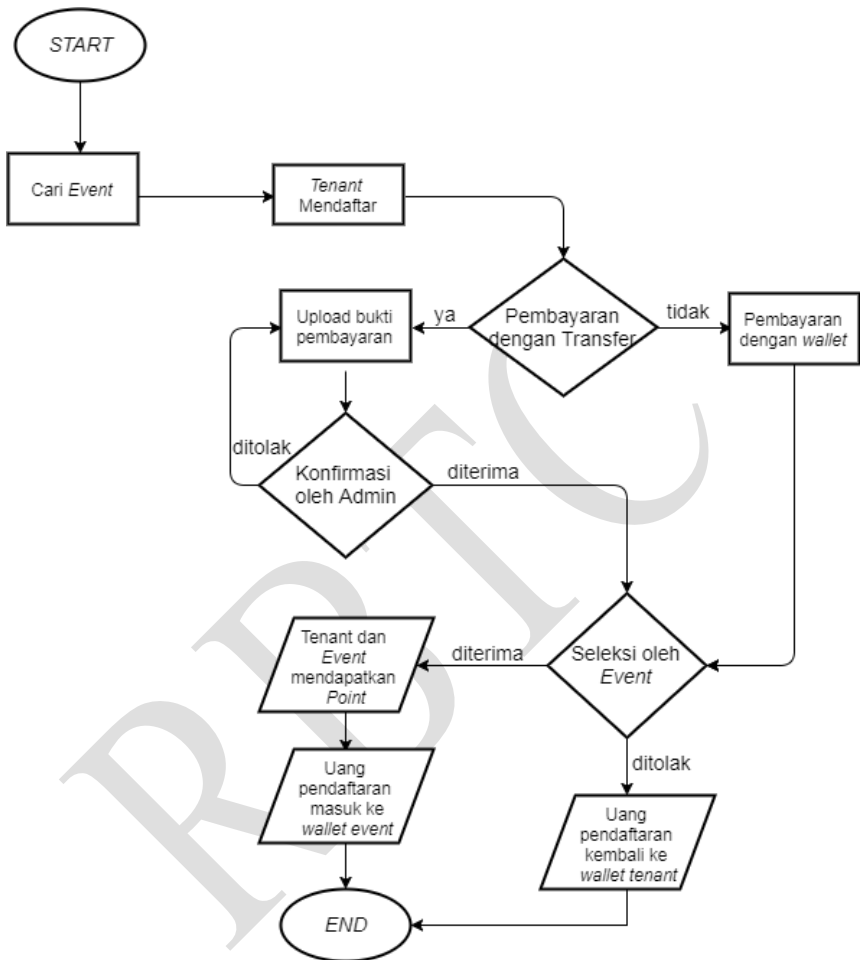
Fitur *Point* merupakan salah satu keuntungan fitur keanggotaan. *Point* ini nantinya dapat ditukar dengan *reward* yang tersedia. Setiap transaksi yang telah selesai terkonfirmasi akan mendapatkan *point* baik bagi *event* maupun *tenant*.

b. *Reward*

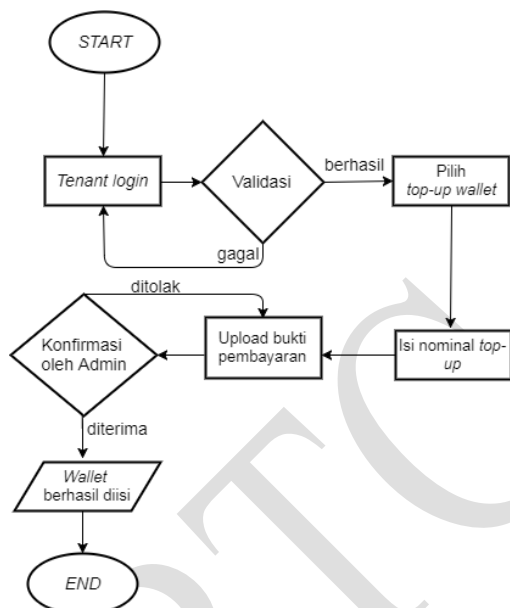
Hadiah atau *Reward* hanya dapat ditukarkan menggunakan *point* yang didapatkan dari transaksi yang pernah dilakukan dalam web CallTenant.

3.1.4 Analisis Proses Bisnis

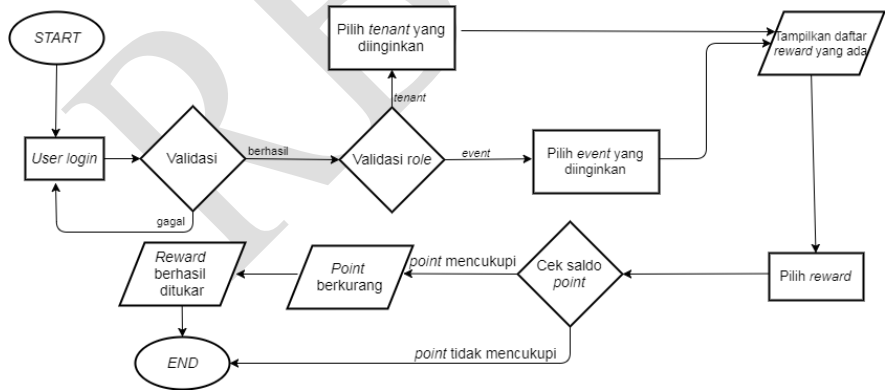
Proses bisnis yang terjadi dalam CallTenant melibatkan *event* dan *tenant* yang terdapat pada bagian transaksi utama antar *event* dan *tenant*, pengisian wallet serta penukaran *reward* dengan *point* pada Gambar 3.1 hingga Gambar 3.3.



Gambar 3.1 Alur Transaksi Utama dalam CallTenant



Gambar 3.2 Alur Pengisian *Wallet*



Gambar 3.3 Alur Penukaran *Reward*

3.1.5 Analisis Kebutuhan Sistem

Analisis spesifikasi kebutuhan dalam aplikasi ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses yang dibutuhkan dan dijalankan dalam sistem. Kebutuhan fungsional dihasilkan dari analisis proses bisnis yang telah dijelaskan sebelumnya. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.2.

Tabel 3.2 Kebutuhan Fungsional

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Mengelola data <i>frequently asked questions</i>	Melihat, menambah, mengubah, dan menghapus data <i>Frequently Asked Questions</i>
F-002	Mengelola pelaporan keuangan	Melihat laporan keuangan yang terdiri berdasarkan transaksi pendaftaran dan <i>wallet user</i>
F-003	Membalas pesan <i>contact us</i>	Membalas pesan <i>Contact Us</i> yang berasal dari pengguna umum seperti non-member, tenant, maupun <i>event</i>
F-004	Mengelola data <i>reward</i> dan <i>point</i>	Mengelola halaman <i>reward</i> yang terdiri dari menambah, mengubah, dan menghapus data <i>reward</i>
F-005	Mengelola pelaporan data <i>tenant</i>	Mengelola pelaporan data <i>tenant</i> yang terdiri dari melihat data mengenai <i>tenant</i> serta

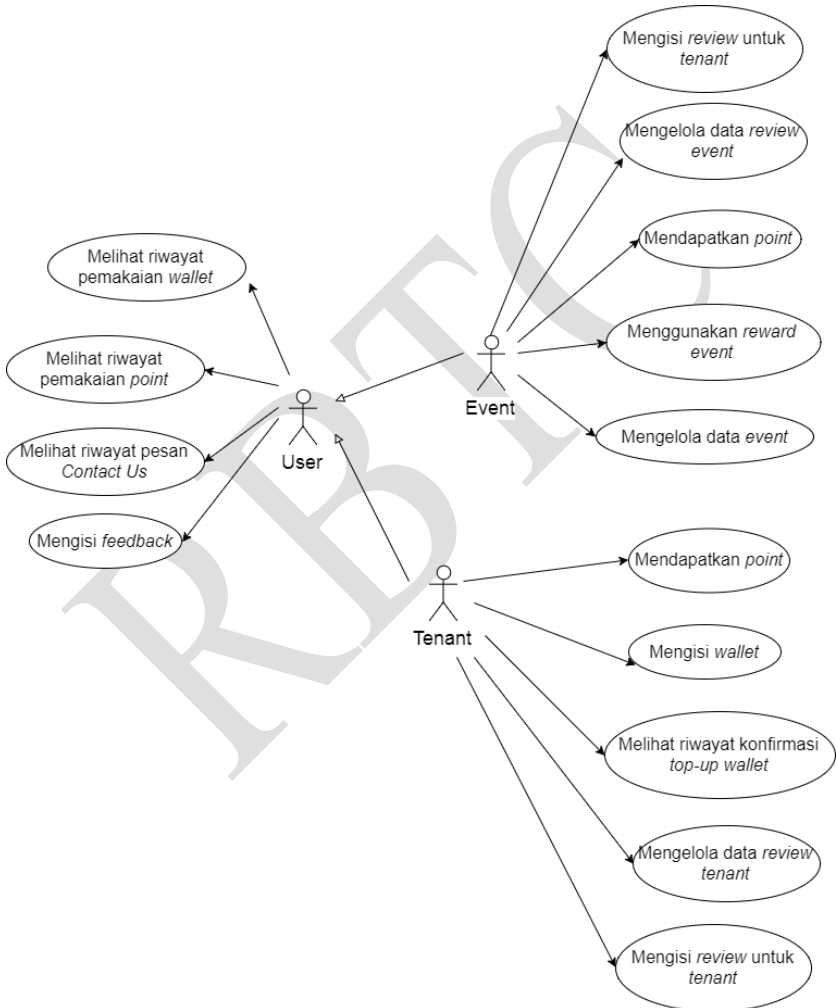
		<i>review</i> yang tertuju bagi <i>tenant</i> tersebut
F-006	Mengelola pelaporan data <i>event</i>	Mengelola pelaporan data <i>event</i> yang terdiri dari melihat data mengenai <i>event</i> serta <i>review</i> yang tertuju bagi <i>event</i> tersebut
F-007	Mengelola pelaporan data <i>user</i>	Mengelola pelaporan data <i>user</i> yang terdiri dari data mengenai <i>user</i>
F-008	Melihat hasil <i>feedback</i>	Melihat hasil pengisian <i>feedback</i> oleh <i>user</i>
F-009	Menggunakan <i>reward event</i>	<i>Event</i> dapat menukarkan <i>point</i> -nya dengan <i>reward</i> yang tersedia
F-010	Menggunakan <i>reward tenant</i>	<i>Tenant</i> dapat menukarkan <i>point</i> -nya dengan <i>reward</i> yang tersedia
F-011	<i>Event</i> mendapatkan <i>point</i>	<i>Event</i> mendapatkan <i>point</i> setelah menyetujui pendaftaran <i>tenant</i>
F-012	<i>Tenant</i> mendapatkan <i>point</i>	<i>Tenant</i> mendapatkan <i>point</i> setelah mendaftar pada <i>event</i> dan pendaftarannya disetujui oleh <i>event</i> tersebut
F-013	Mengisi <i>feedback</i>	<i>User</i> dapat mengisi <i>feedback</i> untuk CallTenant

F-014	Mengisi <i>wallet</i>	<i>Tenant</i> dapat mengisi <i>wallet</i> dengan menyelesaikan pembayaran melalui transfer bank
F-015	Melihat riwayat konfirmasi <i>top-up wallet</i>	<i>Tenant</i> dapat melihat riwayat dan status dari aktivitas <i>top-up wallet</i>
F-016	Melihat riwayat pesan <i>ontact Us</i>	<i>User</i> baik <i>tenant</i> maupun <i>event</i> dapat melihat riwayat pesan dan balasannya yang pernah dikirim melalui fitur <i>Contact Us</i>
F-017	Melihat riwayat pemakaian <i>point</i>	<i>User</i> baik <i>tenant</i> maupun <i>event</i> dapat melihat riwayat penggunaan <i>point</i>
F-018	Melihat riwayat pemakaian <i>wallet</i>	<i>User</i> baik <i>tenant</i> maupun <i>event</i> dapat melihat riwayat penggunaan <i>wallet</i>
F-019	Mengelola data <i>review tenant</i>	Mengelola data <i>review tenant</i> dapat terdiri dari melihat semua <i>review</i> yang tertuju pada <i>tenant</i> tersebut serta membalas <i>review</i> -nya
F-020	Mengelola data <i>review event</i>	Mengelola data <i>review event</i> dapat terdiri dari melihat semua <i>review</i> yang tertuju pada <i>event</i> tersebut serta membalas <i>review</i> -nya

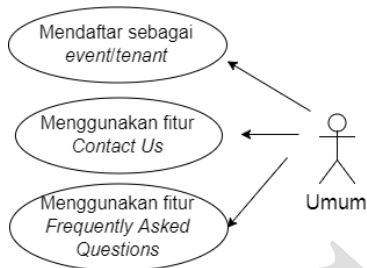
F-021	Menggunakan fitur <i>Contact Us</i>	<i>User</i> baik non-member, tenant maupun event dapat mengirimkan pesan melalui fitur <i>Contact Us</i>
F-022	Menggunakan fitur <i>Frequently Asked Questions</i>	<i>User</i> baik non-member, tenant maupun event dapat melihat data <i>Frequently Asked Questions</i>
F-023	Mengisi <i>review</i> untuk tenant	<i>Event</i> dapat mengisi <i>review</i> untuk tenant yang terdaftar padanya
F-024	Mengisi <i>review</i> untuk event	<i>Tenant</i> dapat mengisi <i>review</i> untuk event yang pernah didaftarkan olehnya
F-025	Mendaftar sebagai <i>event/tenant</i>	Pengguna umum dapat mendaftarkan diri pada CallTenant

3.1.6 Analisis Aktor

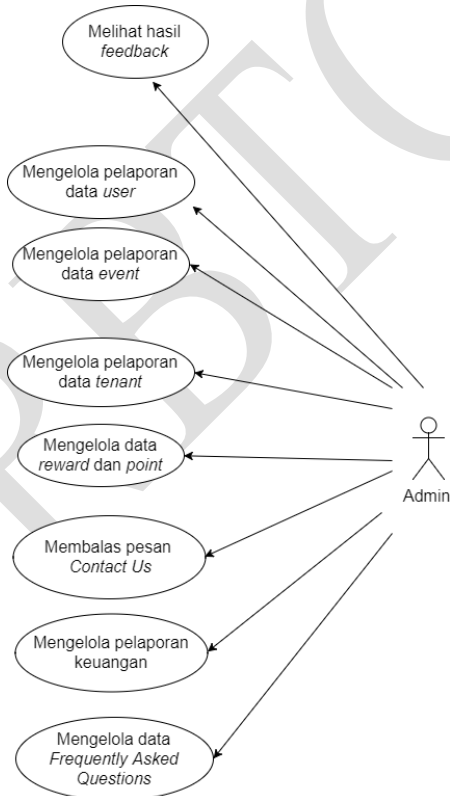
Aktor yang terlibat langsung dalam sistem CallTenant terdapat empat, yaitu Umum, *User* yang digeneralisasi menjadi *Event* dan *Tenant*, serta Admin pada Gambar 3.4 hingga Gambar 3.6.



Gambar 3.4 Use Case pada User (Event dan Tenant)



Gambar 3.5 Use Case pada Pengguna Umum



Gambar 3.6 Use Case pada Admin

3.2 Perancangan Sistem

Tahap perancangan sistem dibagi menjadi beberapa bagian yaitu kasus penggunaan, dan perancangan antarmuka aplikasi.

3.2.1 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan sebelumnya pada spesifikasi kebutuhan fungsional seperti pada Tabel 3.3.

Tabel 3.3 Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Mengelola data <i>Frequently Asked Questions</i>
UC-002	Mengelola pelaporan keuangan
UC-003	Membalas pesan <i>Contact Us</i>
UC-004	Mengelola data <i>reward</i> dan <i>point</i>
UC-005	Mengelola pelaporan data <i>tenant</i>
UC-006	Mengelola pelaporan data <i>event</i>
UC-007	Mengelola pelaporan data <i>user</i>
UC-008	Melihat hasil <i>feedback</i>
UC-009	Menggunakan <i>reward event</i>
UC-010	Menggunakan <i>reward tenant</i>
UC-011	Event mendapatkan <i>point</i>
UC-012	<i>Tenant</i> mendapatkan <i>point</i>
UC-013	Mengisi <i>feedback</i>
UC-014	Mengisi <i>wallet</i>
UC-015	Melihat riwayat konfirmasi <i>top-up wallet</i>
UC-016	Menggunakan <i>reward event</i>
UC-017	Melihat riwayat pemakaian <i>point</i>
UC-018	Melihat riwayat pemakaian <i>wallet</i>
UC-019	Mengelola data <i>review tenant</i>
UC-020	Mengelola data <i>review event</i>
UC-021	Menggunakan fitur <i>Contact Us</i>
UC-022	Menggunakan fitur <i>Frequently Asked Questions</i>

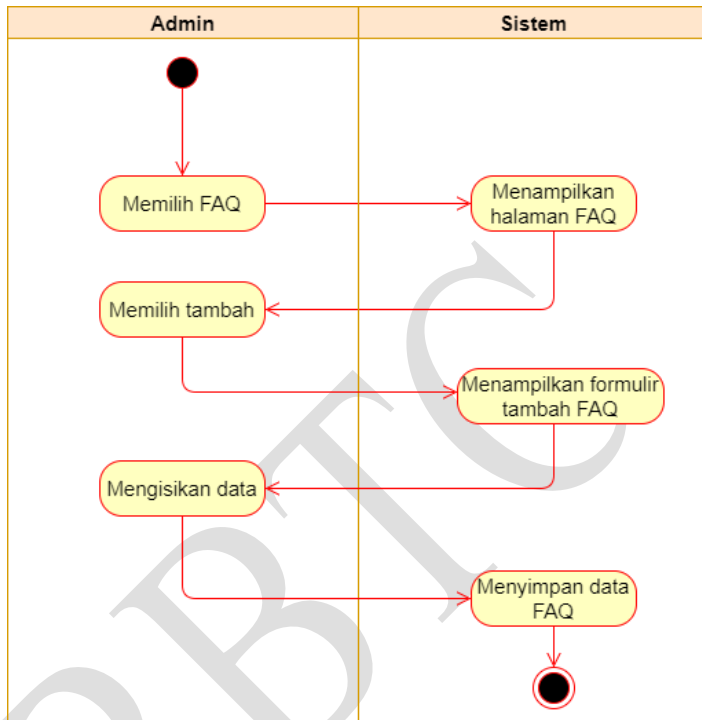
UC-023	Mengisi <i>review</i> untuk <i>tenant</i>
UC-024	Mengisi <i>review</i> untuk <i>event</i>
UC-025	Mendaftar sebagai <i>event/tenant</i>

3.2.1.1 Mengelola Data *Frequently Asked Questions*

Pada kasus penggunaan ini, admin mengelola halaman *FAQ* terdiri dari menambah, mengubah, dan menghapus data *FAQ*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.4 hingga Tabel 3.6 dan diagram aktivitas pada Gambar 3.7 hingga Gambar 3.9.

Tabel 3.4 Menambah Data *Frequently Asked Questions*

Kode	UC-001.1
Nama	Menambah Data <i>Frequently Asked Questions</i>
Aktor	Admin
Deskripsi	Admin menambah data <i>Frequently Asked Questions</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Data <i>Frequently Asked Questions</i> berhasil ditambahkan
Alur Kejadian Normal	
1. Admin memilih menu <i>FAQ</i> 2. Sistem menampilkan halaman <i>FAQ</i> 3. Admin memilih tambah data 4. Sistem menampilkan formulir tambah <i>FAQ</i> baru 5. Admin mengisikan data 6. Sistem menyimpan data <i>FAQ</i>	
Alur Kejadian Alternatif	
-	

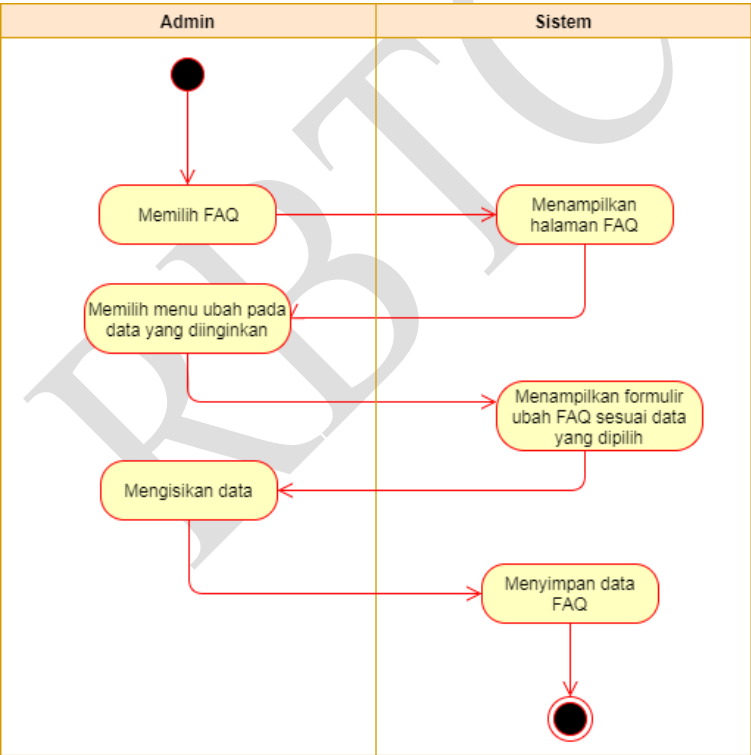


Gambar 3.7 Diagram Aktivitas Menambah Data *Frequently Asked Questions*

Tabel 3.5 Mengubah Data *Frequently Asked Questions*

Kode	UC-001.2
Nama	Mengubah Data <i>Frequently Asked Questions</i>
Aktor	Admin
Deskripsi	Admin dapat mengubah data <i>Frequently Asked Questions</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Data <i>Frequently Asked Questions</i> berhasil diubah

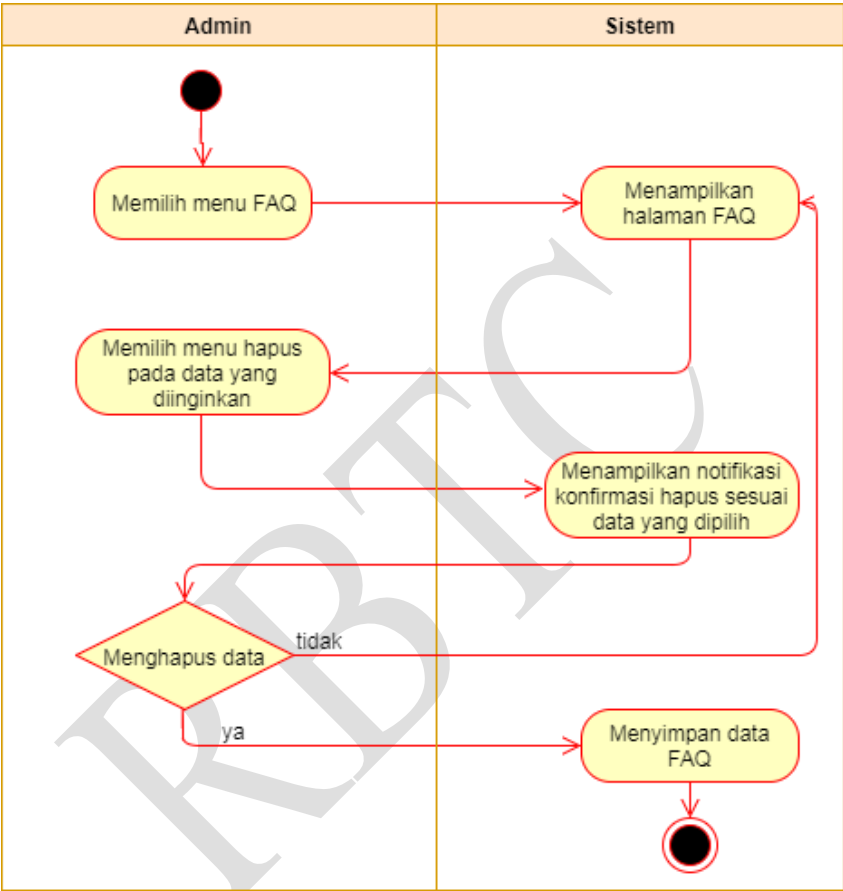
Alur Kejadian Normal
1. Admin memilih menu FAQ
2. Sistem menampilkan halaman FAQ
3. Admin memilih menu ubah pada data yang diinginkan
4. Sistem menampilkan formulir ubah FAQ sesuai yang dipilih
5. Admin mengisikan data
6. Sistem menyimpan data FAQ
Alur Kejadian Alternatif
-



Gambar 3.8 Diagram Aktivitas Mengubah Data *Frequently Asked Questions*

Tabel 3.6 Menghapus Data *Frequently Asked Questions*

Kode	UC-001.3
Nama	Menghapus Data <i>Frequently Asked Questions</i>
Aktor	Admin
Deskripsi	Admin dapat menghapus data <i>Frequently Asked Questions</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Data <i>Frequently Asked Questions</i> berhasil dihapus
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu FAQ2. Sistem menampilkan halaman FAQ3. Admin memilih menu hapus pada data yang diinginkan4. Sistem menampilkan notifikasi konfirmasi hapus sesuai data yang dipilih5. Admin menghapus data6. Sistem menyimpan data FAQ	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">5. Admin tidak menghapus data5.1 Kembali ke nomor 2	



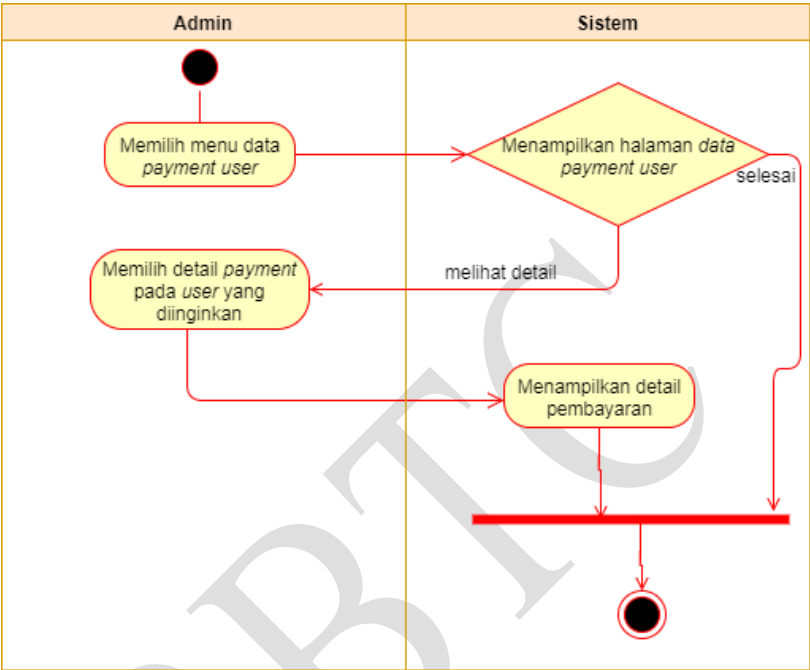
Gambar 3.9 Diagram Aktivitas Menghapus Data *Frequently Asked Questions*

3.2.1.2 Mengelola Pelaporan Keuangan

Pada kasus penggunaan ini, admin dapat melihat laporan keuangan yang terdiri berdasarkan transaksi pendaftaran dan pengisian *wallet user*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.7 hingga Tabel 3.10 dan diagram aktivitas pada Gambar 3.10 hingga Gambar 3.13.

Tabel 3.7 Melihat Laporan Keuangan dari Transaksi Pendaftaran

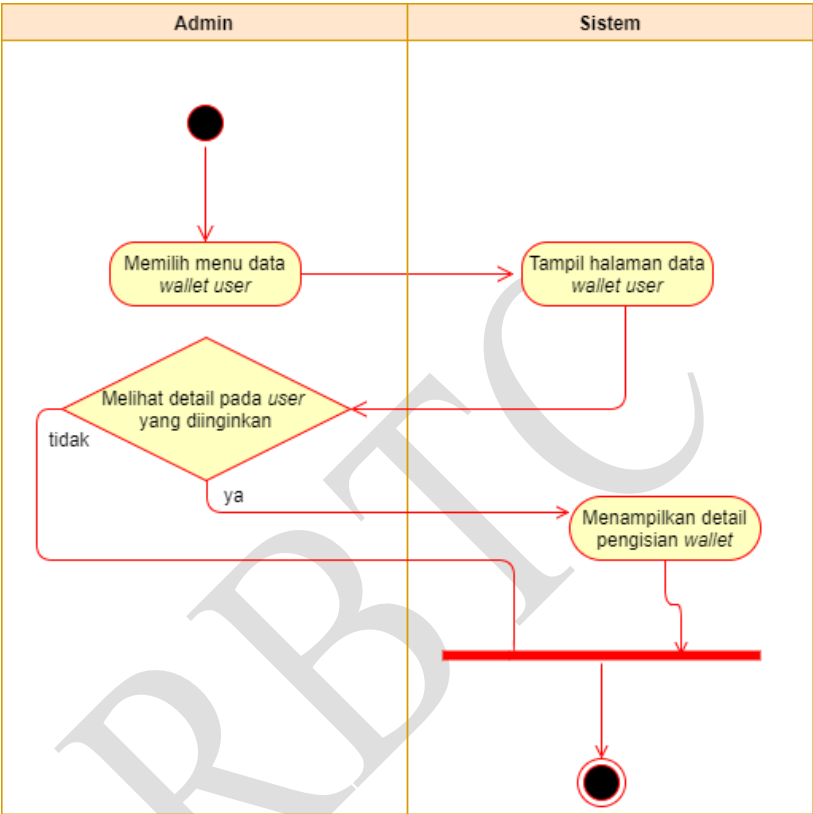
Kode	UC-002.1
Nama	Melihat Laporan Keuangan dari Transaksi Pendaftaran
Aktor	Admin
Deskripsi	Admin dapat melihat laporan keuangan yang berasal dari transaksi pendaftaran antar <i>event</i> dan <i>tenant</i>
Kondisi Awal	Admin berada di halaman <i>dashboard admin</i>
Kondisi akhir	Admin berhasil melihat laporan keuangan dari transaksi pendaftaran antar <i>event</i> dan <i>tenant</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Data Payment User</i> pada menu <i>Payment</i> dalam <i>Financial Report</i>2. Sistem menampilkan halaman <i>Data Payment User</i>3. Admin memilih detail <i>payment</i> pada data <i>user</i> yang diinginkan4. Sistem menampilkan detail pembayaran dari <i>user</i> tersebut	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">3. Admin tidak melihat detail <i>payment</i> <p>3.1 Kembali ke nomor 4</p>	



Gambar 3.10 Diagram Aktivitas Melihat Laporan Keuangan dari Transaksi Pendaftaran

Tabel 3.8 Melihat Laporan Keuangan dari Pengisian *Wallet User*

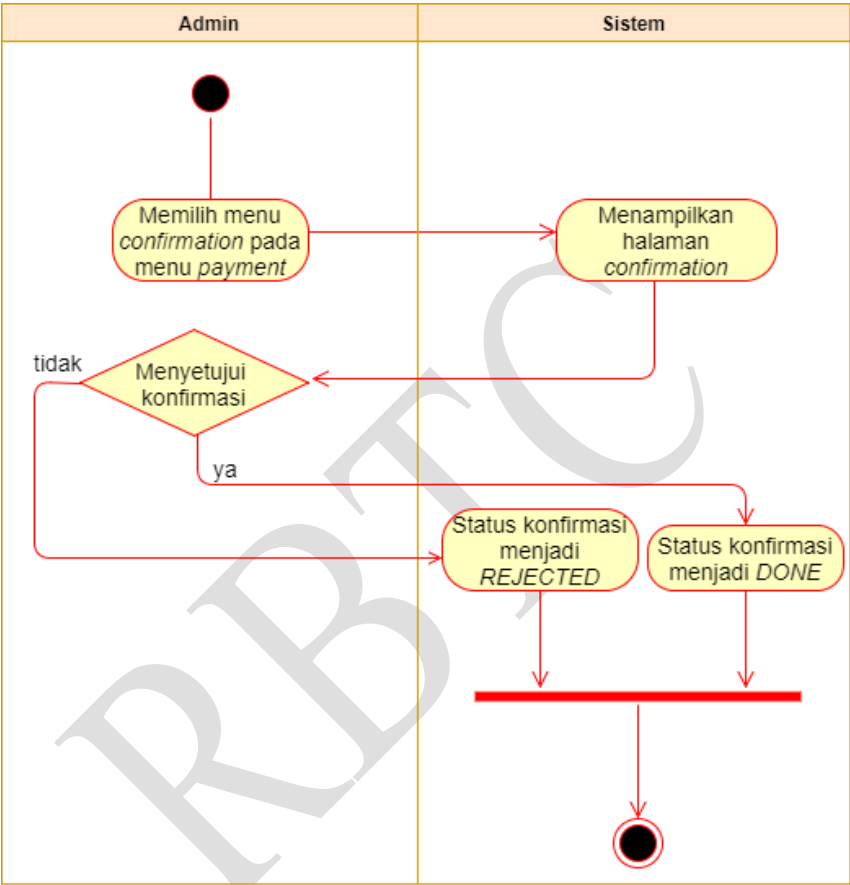
Kode	UC-002.2
Nama	Melihat Laporan Keuangan dari Pengisian <i>Wallet User</i>
Aktor	Admin
Deskripsi	Admin dapat melihat laporan keuangan yang berasal dari pengisian <i>wallet tenant</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat laporan keuangan dari pengisian <i>wallet tenant</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu Data <i>Wallet User</i> pada menu <i>Wallet</i> dalam <i>Financial Report</i>2. Sistem menampilkan halaman Data <i>Wallet User</i>3. Admin memilih detail <i>wallet</i> pada data <i>user</i> yang diinginkan4. Sistem menampilkan detail pengisian <i>wallet</i> dari <i>user</i> tersebut	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">3. Admin tidak memilih detail <i>wallet</i> pada <i>user</i> yang diinginkan <p>3.1 Kembali ke nomor 4</p>	



Gambar 3.11 Diagram Aktivitas Melihat Laporan Keuangan dari Pengisian *Wallet User*

Tabel 3.9 Mengonfirmasi Pembayaran Transaksi Pendaftaran

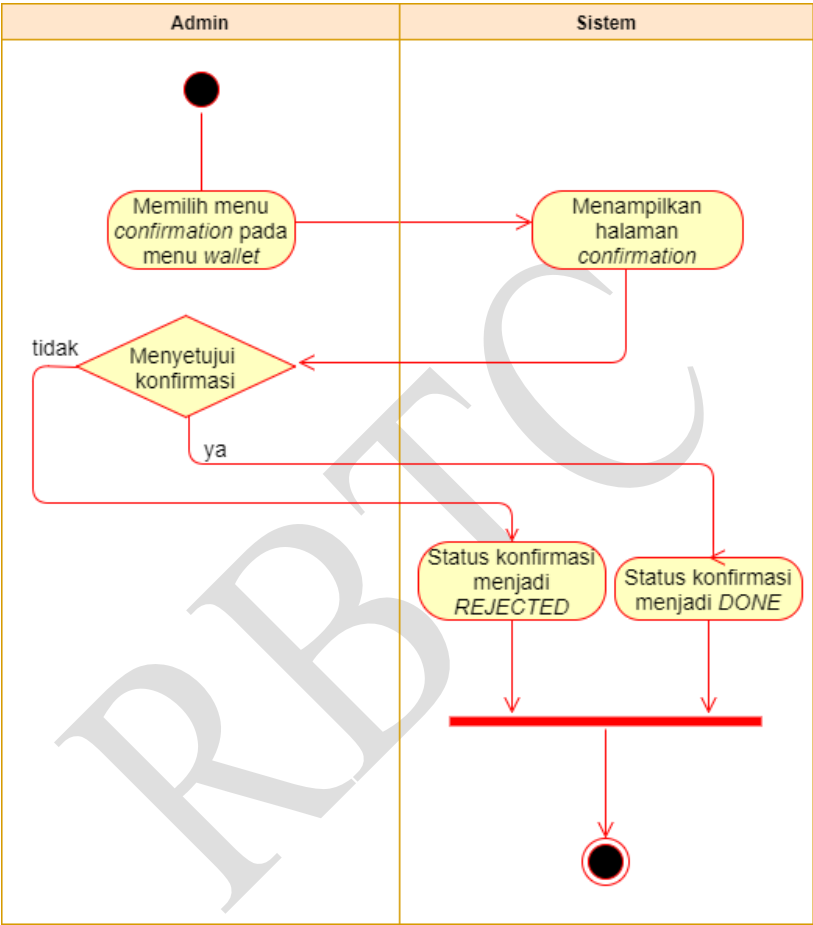
Kode	UC-002.3
Nama	Mengonfirmasi Pembayaran Transaksi Pendaftaran
Aktor	Admin
Deskripsi	Admin dapat menyetujui atau menolak pembayaran transaksi pendaftaran dari <i>tenant</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil mengonfirmasi pembayaran transaksi pendaftaran
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Confirmation</i> pada menu <i>Payment</i> dalam <i>Financial Report</i>2. Sistem menampilkan halaman <i>Confirmation</i>3. Admin menyetujui data yang memiliki status <i>PENDING</i>4. Sistem menyimpan data dan status konfirmasi berubah menjadi <i>DONE</i>	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">3. Admin tidak menyetujui data yang memiliki status <i>PENDING</i><ol style="list-style-type: none">3.1 Sistem menyimpan data dan status konfirmasi berubah menjadi <i>REJECTED</i>	



Gambar 3.12 Diagram Aktivitas Mengonfirmasi Pembayaran Transaksi Pendaftaran

Tabel 3.10 Mengonfirmasi Pembayaran Pengisian *Wallet*

Kode	UC-002.4
Nama	Mengonfirmasi Pembayaran Pengisian <i>Wallet</i>
Aktor	Admin
Deskripsi	Admin dapat menyetujui atau menolak pembayaran pengisian <i>wallet</i> dari <i>tenant</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil mengonfirmasi pembayaran pengisian <i>wallet</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Confirmation</i> pada menu <i>Wallet</i> dalam <i>Financial Report</i>2. Sistem menampilkan halaman <i>Confirmation</i>3. Admin menyetujui data yang memiliki status <i>PENDING</i>4. Sistem menyimpan data dan status konfirmasi berubah menjadi <i>DONE</i>	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">3. Admin tidak menyetujui data yang memiliki status <i>PENDING</i><ol style="list-style-type: none">3.1 Sistem menyimpan data dan status konfirmasi berubah menjadi <i>REJECTED</i>	



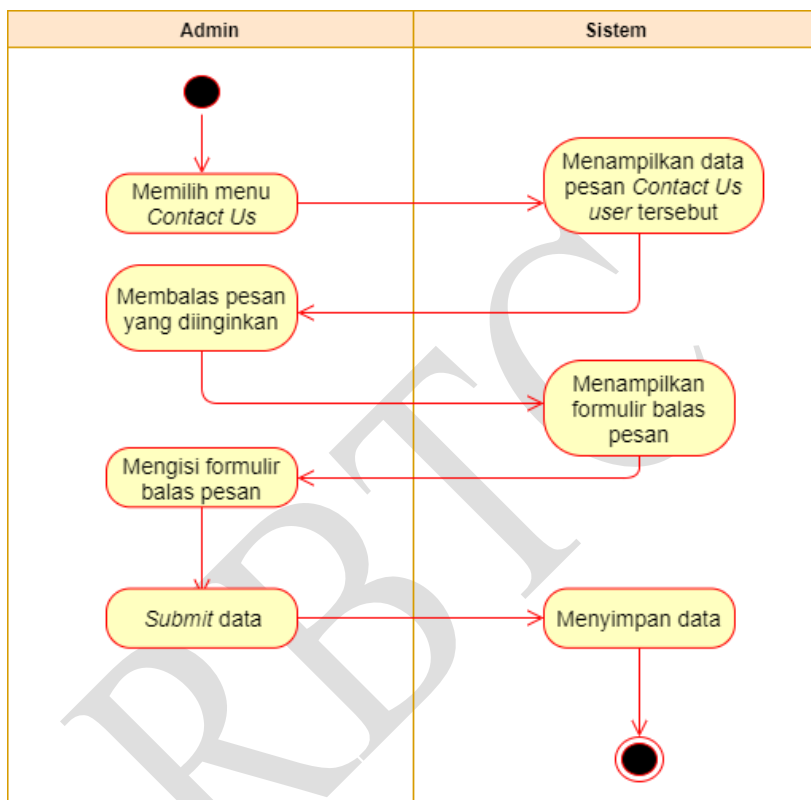
Gambar 3.13 Diagram Aktivitas Mengonfirmasi Pembayaran Pengisian *Wallet*

3.2.1.3 Membalas Pesan *Contact Us*

Pada kasus penggunaan ini, admin dapat membalas pesan *Contact Us* yang berasal dari pengguna umum seperti *non-member*, *tenant*, maupun *event*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.11 dan diagram aktivitas pada Gambar 3.14.

Tabel 3.11 Membalas Pesan *Contact Us*

Kode	UC-003
Nama	Membalas Pesan dari Fitur <i>Contact Us</i>
Aktor	Admin
Deskripsi	Admin membalas pesan <i>Contact Us</i> dari <i>user</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Pesan <i>user</i> berhasil dibalas
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Contact Us</i>2. Sistem menampilkan data pesan <i>Contact Us</i> dari <i>user</i>3. Admin membalas pesan yang diinginkan4. Sistem menampilkan formulir balas pesan <i>user</i> yang dipilih5. Admin mengisi formulir balas pesan6. Admin meng-<i>submit</i> data7. Sistem menyimpan data	
Alur Kejadian Alternatif	
-	



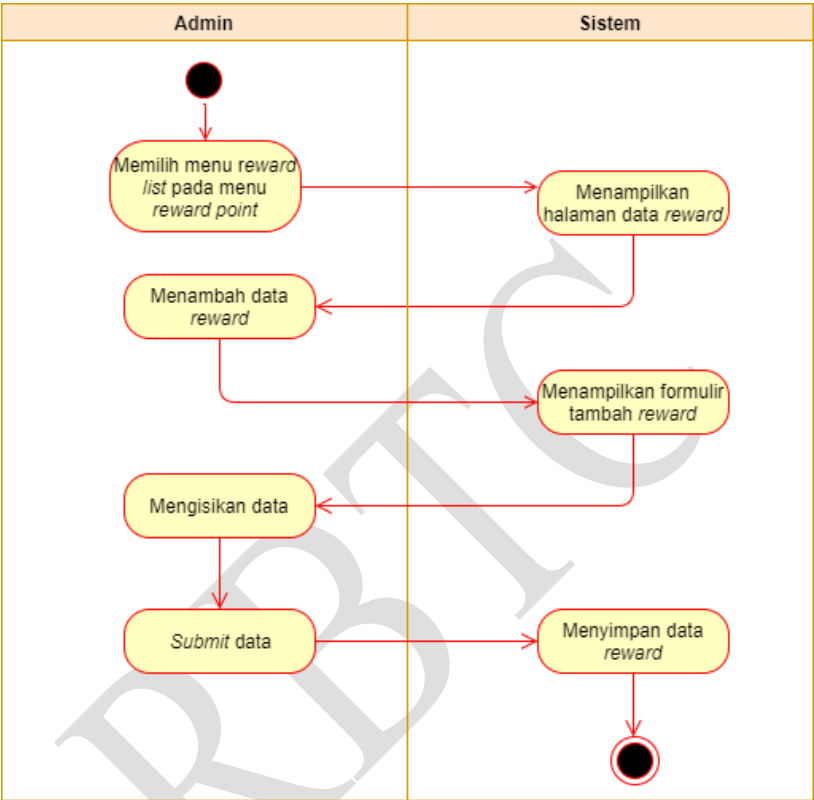
Gambar 3.14 Diagram Aktivitas Membalas Pesan *Contact Us*

3.2.1.4 Mengelola Data *Reward dan Point*

Pada kasus penggunaan ini, admin mengelola halaman *reward* yang terdiri dari menambah, mengubah, dan menghapus data *reward*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.12 hingga Tabel 3.17 dan diagram aktivitas pada Gambar 3.15 hingga Gambar 3.20.

Tabel 3.12 Menambah Data *Reward*

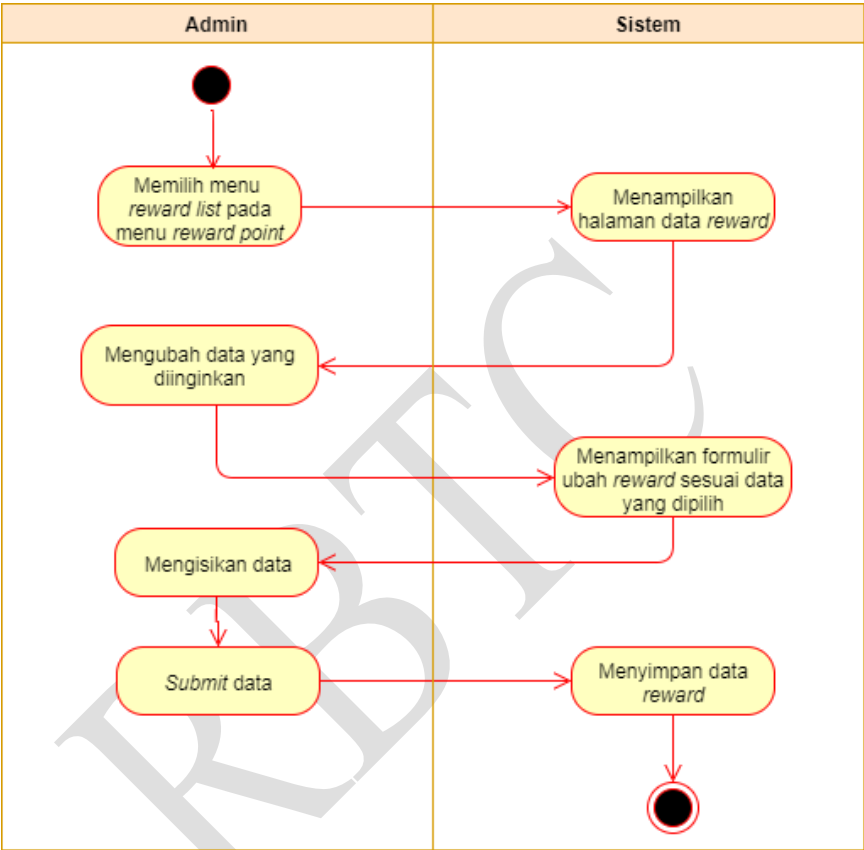
Kode	UC-004.1
Nama	Menambah Data <i>Reward</i>
Aktor	Admin
Deskripsi	Admin menambah data <i>Reward</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	<i>Reward</i> berhasil ditambahkan ke dalam basis data
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>reward list</i> pada menu <i>reward point</i>2. Sistem menampilkan halaman data <i>reward</i>3. Admin menambah data <i>reward</i>4. Sistem menampilkan formulir tambah data <i>reward</i>5. Admin mengisi data <i>reward</i>6. Admin meng-<i>submit</i> data7. Sistem berhasil menyimpan data	
Alur Kejadian Alternatif	
-	



Gambar 3.15 Diagram Aktivitas Menambah Data *Reward*

Tabel 3.13 Mengubah Data *Reward*

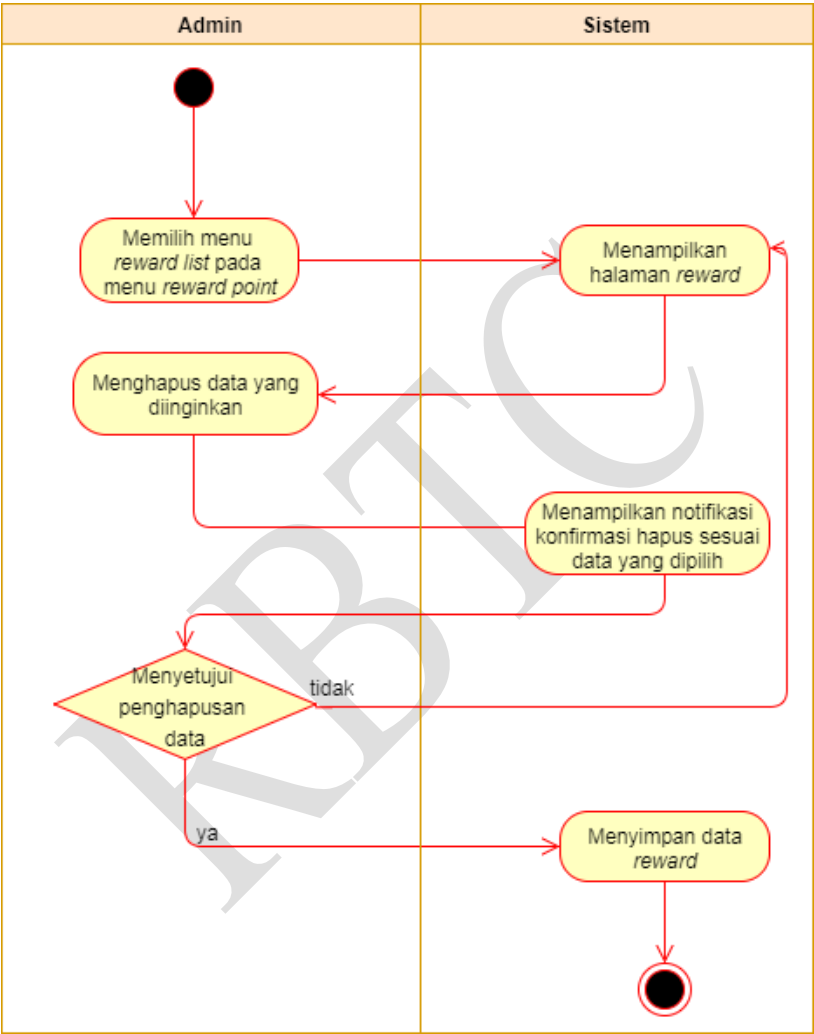
Kode	UC-004.2
Nama	Mengubah Data <i>Reward</i>
Aktor	Admin
Deskripsi	Admin dapat mengubah data <i>Reward</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Data <i>Reward</i> berhasil diubah
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>reward list</i> pada menu <i>reward point</i>2. Sistem menampilkan halaman data <i>reward</i>3. Admin mengubah data yang diinginkan4. Sistem menampilkan formulir ubah <i>Reward</i> sesuai yang dipilih5. Admin mengisikan data6. Admin meng-<i>submit</i> data7. Sistem menyimpan data <i>reward</i>	
Alur Kejadian Alternatif	
-	



Gambar 3.16 Diagram Aktivitas Mengubah Data *Reward*

Tabel 3.14 Menghapus Data *Reward*

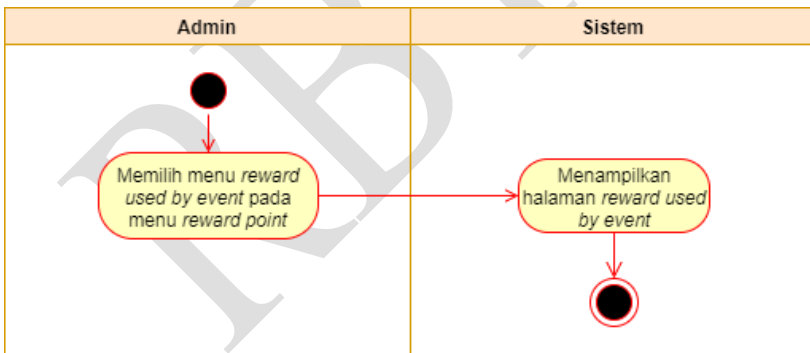
Kode	UC-004.3
Nama	Menghapus Data <i>Reward</i>
Aktor	Admin
Deskripsi	Admin dapat menghapus data <i>Reward</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Data <i>Reward</i> berhasil dihapus
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>reward list</i> pada menu <i>reward point</i>2. Sistem menampilkan halaman <i>Reward</i>3. Admin menghapus data yang diinginkan4. Sistem menampilkan notifikasi konfirmasi hapus sesuai data yang dipilih5. Admin menyetujui penghapusan data6. Sistem menyimpan data <i>Reward</i>	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">5.Admin tidak menyetujui penghapusan data5.1 Kembali ke nomor 2	



Gambar 3.17 Diagram Aktivitas Menghapus Data *Reward*

Tabel 3.15 Melihat Data *Reward* yang Dipakai *Event*

Kode	UC-004.4
Nama	Melihat Data <i>Reward</i> yang Dipakai <i>Event</i>
Aktor	Admin
Deskripsi	Admin dapat melihat daftar <i>Reward</i> yang sedang atau sudah dipakai oleh <i>Event</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat <i>Reward</i> yang Dipakai <i>Event</i>
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Admin memilih menu <i>Reward Used by Event</i> pada menu <i>Reward Point</i> 2. Sistem menampilkan halaman berisi data <i>Reward Used by Event</i> 	
Alur Kejadian Alternatif	
-	

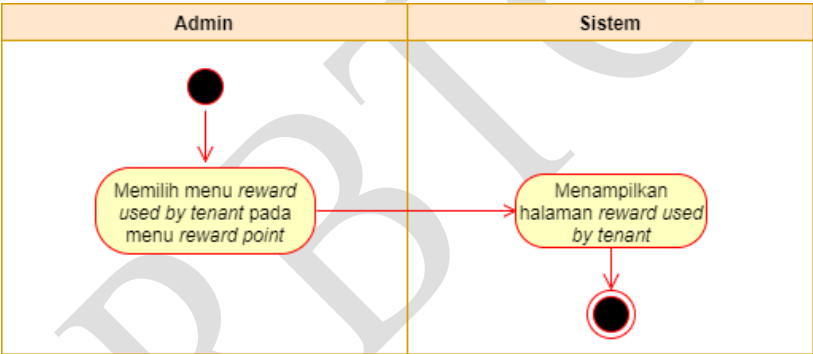


Gambar 3.18 Diagram Aktivitas Melihat Data *Reward* yang Dipakai *Event*

Tabel 3.16 Melihat Data *Reward* yang Dipakai *Tenant*

Kode	UC-004.5
Nama	Melihat Data <i>Reward</i> yang Dipakai <i>Tenant</i>
Aktor	Admin

Deskripsi	Admin dapat melihat daftar <i>Reward</i> yang sedang atau sudah dipakai oleh <i>Tenant</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat <i>Reward</i> yang Dipakai <i>Tenant</i>
Alur Kejadian Normal	
1. Admin memilih menu <i>Reward Used by Tenant</i> pada menu <i>Reward</i> 2. Sistem menampilkan halaman berisi data <i>Reward Used by Tenant</i>	
Alur Kejadian Alternatif	
-	

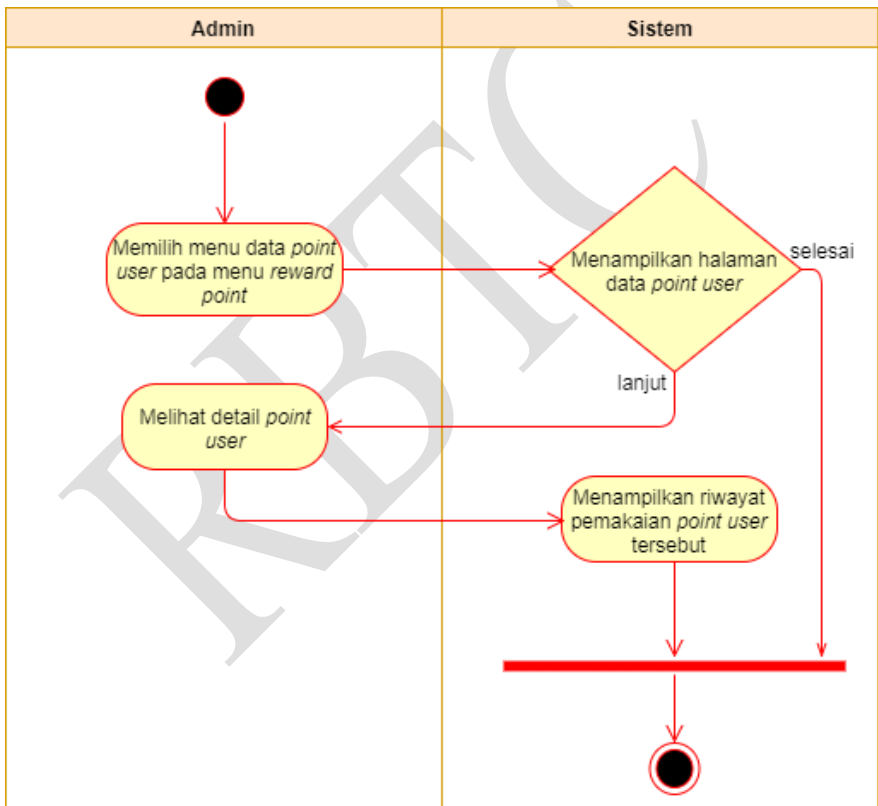


Gambar 3.19 Diagram Aktivitas Melihat Data *Reward* yang Dipakai *Tenant*

Tabel 3.17 Melihat Data *Point User*

Kode	UC-004.6
Nama	Melihat Data <i>Point User</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data dan riwayat pemakaian <i>point user</i>
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>point user</i>
Alur Kejadian Normal	

1. Admin memilih menu Data <i>Point</i> User pada menu <i>Reward Point</i>
2. Sistem menampilkan halaman berisi data <i>point user</i>
Alur Kejadian Alternatif
2. Sistem menampilkan halaman berisi data <i>point user</i>
2.1 Admin melihat detail pada <i>user</i> yang diinginkan
2.2 Sistem menampilkan riwayat pemakaian <i>point</i> <i>user</i> tersebut



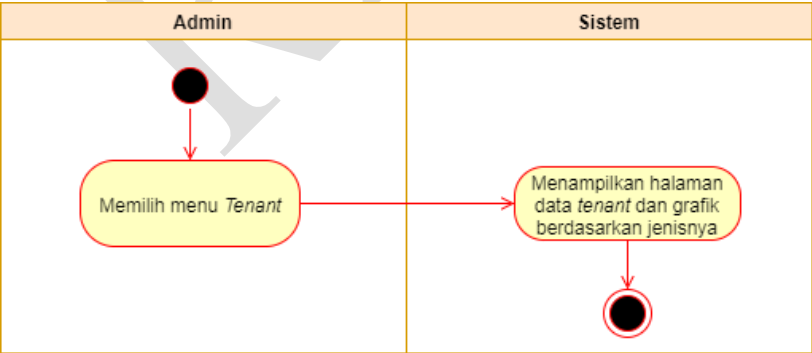
Gambar 3.20 Diagram Aktivitas Melihat Data *Point* User

3.2.1.5 Mengelola Pelaporan Data *Tenant*

Pada kasus penggunaan ini, admin dapat mengelola pelaporan data *tenant* yang terdiri melihat data mengenai *tenant* serta *review* yang tertuju bagi *tenant* tersebut. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.18 dan Tabel 3.19 serta aktivitas diagram pada Gambar 3.21 dan Gambar 3.22.

Tabel 3.18 Melihat Data *Tenant*

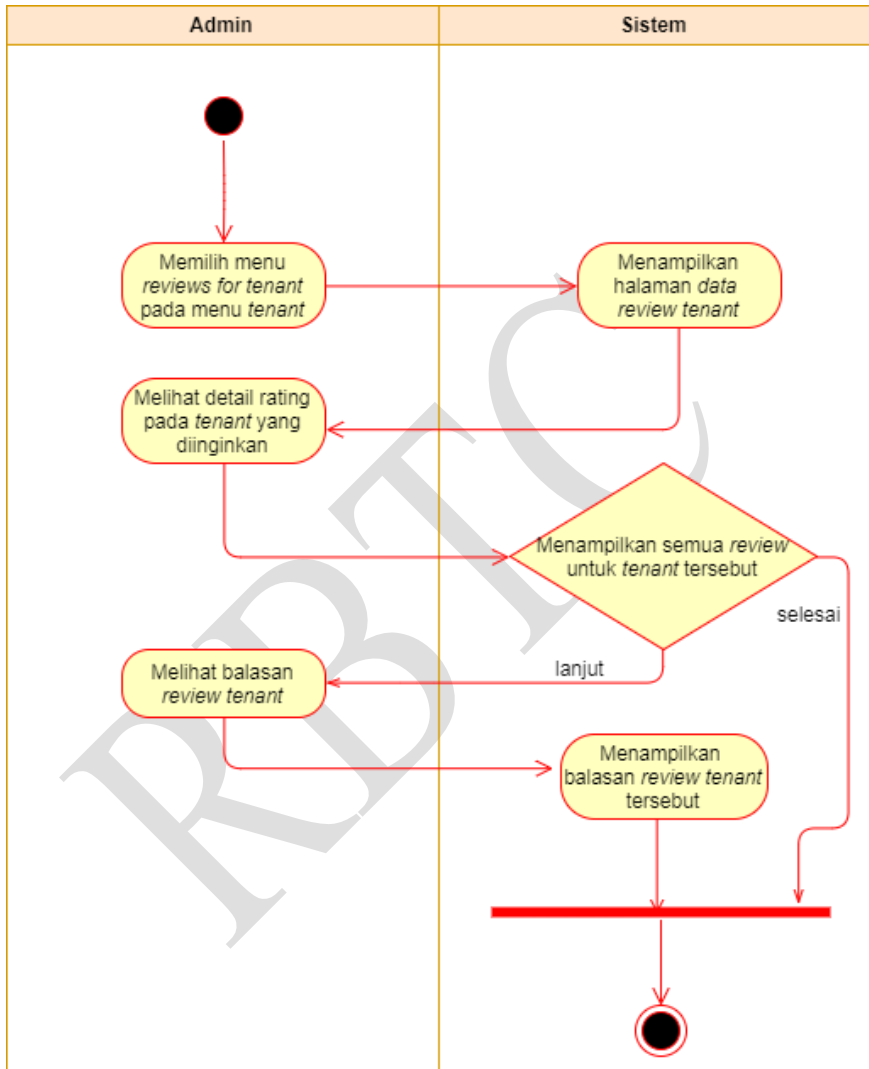
Kode	UC-005.1
Nama	Melihat Data <i>Tenant</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data mengenai <i>tenant</i> serta grafik pendaftar <i>tenant</i> berdasarkan jenisnya
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>tenant</i>
Alur Kejadian Normal	
1. Admin memilih menu <i>Tenant</i> 2. Sistem menampilkan halaman Data <i>Tenant</i> serta grafik jumlahnya berdasarkan jenis <i>tenant</i>	
Alur Kejadian Alternatif	
-	



Gambar 3.21 Diagram Aktivitas Melihat Data *Tenant*

Tabel 3.19 Melihat Data *Review Tenant*

Kode	UC-005.2
Nama	Melihat Data <i>Review Tenant</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data berisi semua <i>review</i> yang tertuju pada <i>tenant</i> tersebut
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>review tenant</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Reviews for Tenant</i> pada menu <i>Tenant</i>2. Sistem menampilkan halaman Data <i>Review Tenant</i>3. Admin melihat detail <i>rating</i> pada <i>tenant</i> yang diinginkan4. Sistem menampilkan semua data <i>review</i> yang tertuju pada <i>tenant</i> tersebut	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">4. Sistem menampilkan semua data <i>review</i> yang tertuju pada <i>tenant</i> tersebut<ol style="list-style-type: none">4.1 Admin melihat balasan <i>review</i> dari <i>tenant</i> tersebut4.2 Admin melihat isi balasan pada <i>review</i> yang diinginkan4.3 Sistem menampilkan balasan dari <i>tenant</i> tersebut	



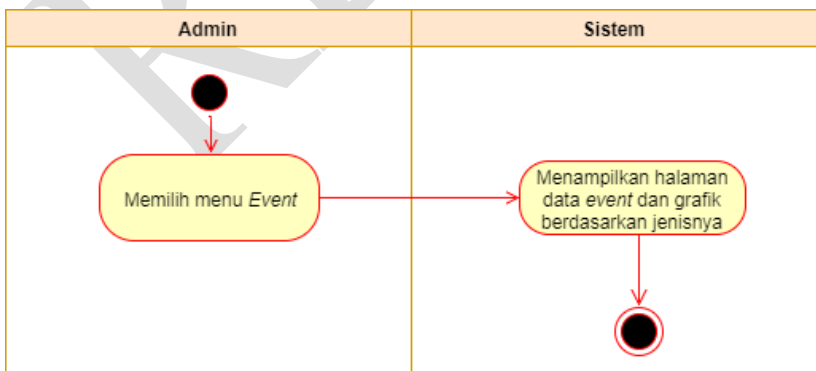
Gambar 3.22 Diagram Aktivitas Melihat Data *Review Tenant*

3.2.1.6 Mengelola Pelaporan Data *Event*

Pada kasus penggunaan ini, admin dapat mengelola pelaporan data *event* yang terdiri melihat data mengenai *event* serta *review* yang tertuju bagi *event* tersebut. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.20 dan Tabel 3.21 serta aktivitas diagram pada Gambar 3.23 dan Gambar 3.24.

Tabel 3.20 Melihat Data *Event*

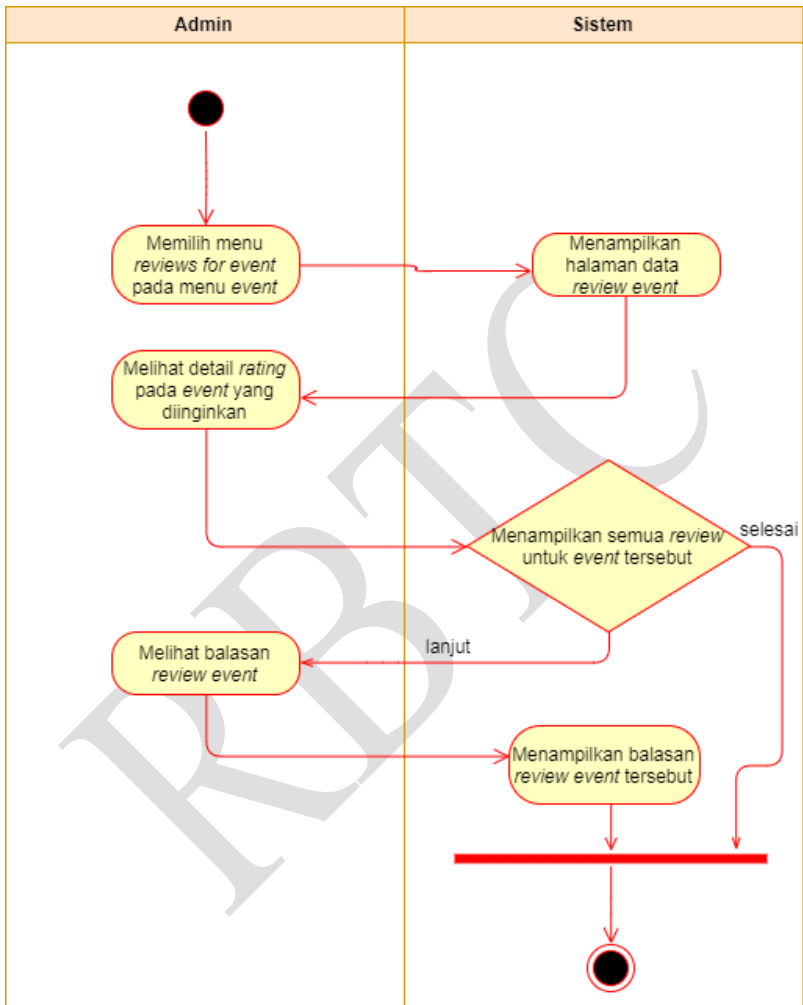
Kode	UC-006.1
Nama	Melihat Data <i>Event</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data mengenai <i>event</i> serta grafik pendaftar <i>event</i> berdasarkan jenisnya
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>event</i>
Alur Kejadian Normal	
1. Admin memilih menu <i>Event</i> 2. Sistem menampilkan halaman Data <i>Event</i> serta grafik jumlahnya berdasarkan jenis <i>event</i>	
Alur Kejadian Alternatif	
-	



Gambar 3.23 Diagram Aktivitas Melihat Data *Event*

Tabel 3.21 Melihat Data Review Event

Kode	UC-006.2
Nama	Melihat Data <i>Review Event</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data berisi semua review yang tertuju pada <i>event</i> tersebut
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>review event</i>
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Admin memilih menu <i>Reviews for Events</i> pada menu <i>Event</i> 2. Sistem menampilkan halaman Data <i>Review Event</i> 3. Admin melihat detail <i>rating</i> pada <i>event</i> yang diinginkan 4. Sistem menampilkan semua data <i>review</i> yang tertuju pada <i>event</i> tersebut 	
Alur Kejadian Alternatif	
<ol style="list-style-type: none"> 4. Sistem menampilkan semua data <i>review</i> yang tertuju pada <i>event</i> tersebut <ol style="list-style-type: none"> 4.1 Admin melihat balasan <i>review</i> dari <i>event</i> tersebut 4.2 Admin melihat isi balasan pada <i>review</i> yang diinginkan 4.3 Sistem menampilkan balasan dari <i>event</i> tersebut 	



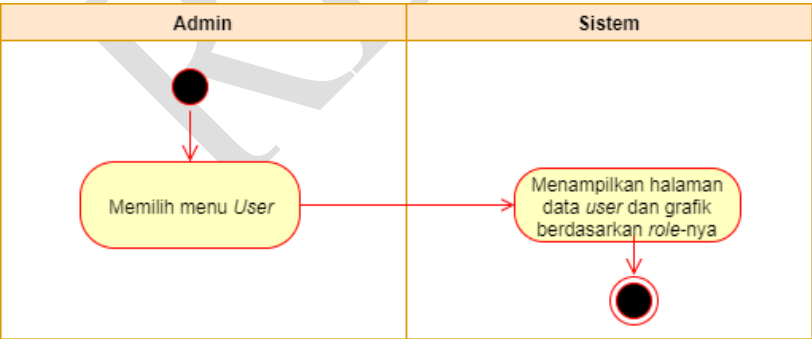
Gambar 3.24 Diagram Aktivitas Melihat Data *Review Event*

3.2.1.7 Mengelola Pelaporan Data User

Pada kasus penggunaan ini, admin dapat mengelola pelaporan data user yang terdiri melihat data mengenai *user*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.22 dan diagram aktivitas pada Gambar 3.25.

Tabel 3.22 Melihat Data *User*

Kode	UC-007
Nama	Melihat Data <i>User</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data mengenai <i>user</i> serta grafik pendaftar <i>user</i> berdasarkan <i>role</i> -nya
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat data <i>user</i>
Alur Kejadian Normal	
1. Admin memilih menu <i>User</i> 2. Sistem menampilkan halaman data user serta grafik jumlahnya berdasarkan jenis <i>role</i> -nya	
Alur Kejadian Alternatif	
-	



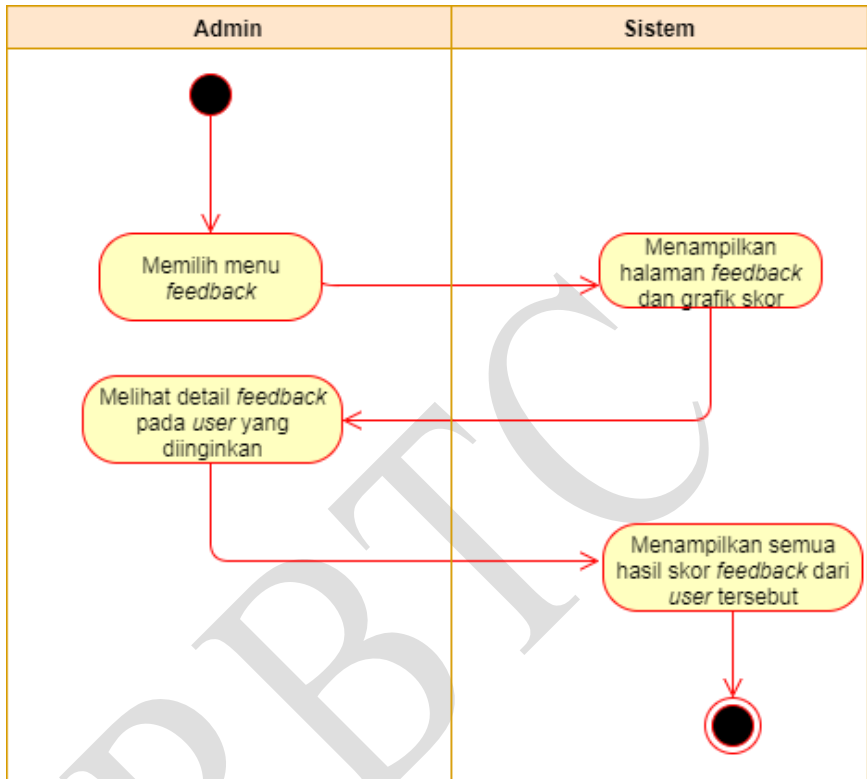
Gambar 3.25 Diagram Aktivitas Melihat Data *User*

3.2.1.8 Melihat Hasil *Feedback*

Pada kasus penggunaan ini, admin dapat melihat hasil *feedback*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.23 dan diagram aktivitas pada Gambar 3.26.

Tabel 3.23 Melihat Hasil *Feedback*

Kode	UC-008
Nama	Melihat Hasil <i>Feedback</i>
Aktor	Admin
Deskripsi	Admin dapat melihat data <i>feedback</i> yang berasal dari <i>user</i> serta grafik skor pada setiap pertanyaannya
Kondisi Awal	Admin berada di halaman <i>dashboard</i> admin
Kondisi akhir	Admin berhasil melihat hasil <i>feedback</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Admin memilih menu <i>Feedback</i>2. Sistem menampilkan halaman berisi data <i>feedback</i> serta grafik skor setiap pertanyaan3. Admin melihat detail <i>feedback</i> pada <i>user</i> yang diinginkan4. Sistem menampilkan semua hasil skor <i>feedback</i> yang diisi oleh <i>user</i> tersebut	
Alur Kejadian Alternatif	
-	



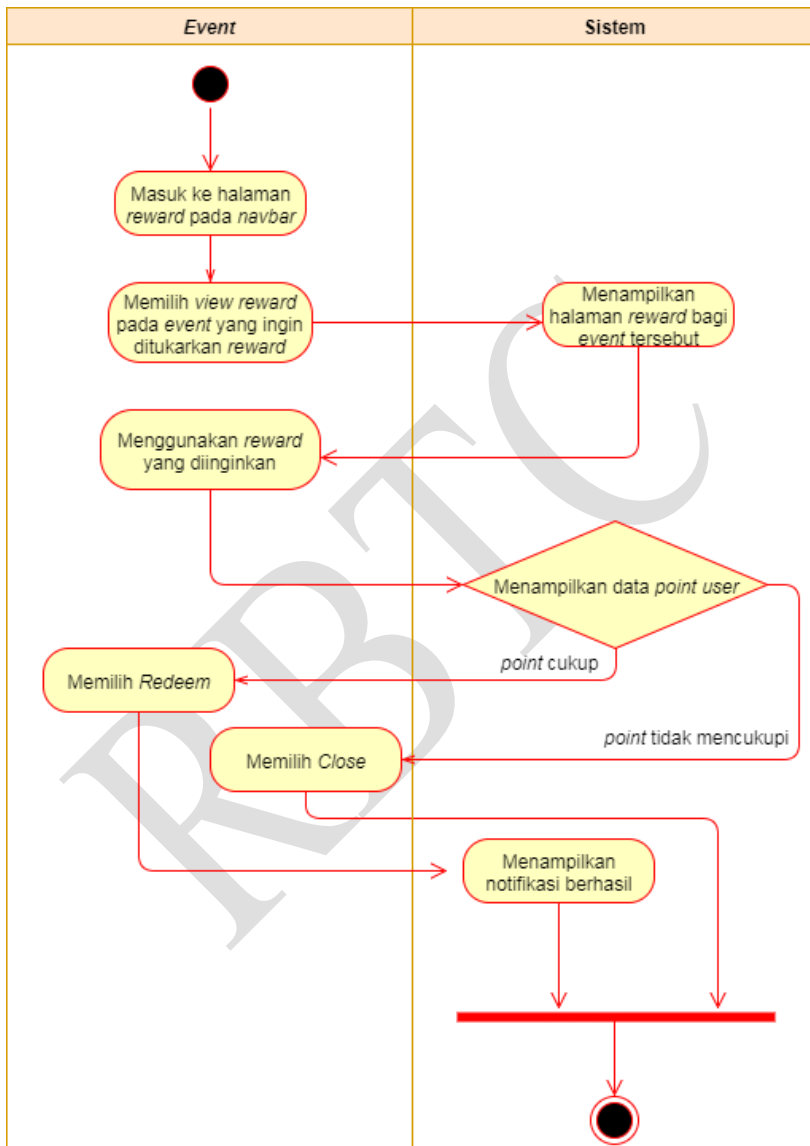
Gambar 3.26 Diagram Aktivitas Melihat Hasil *Feedback*

3.2.1.9 Menggunakan *Reward Event*

Pada kasus penggunaan ini, *event* dapat menukarkan *point*-nya dengan *reward* yang tersedia. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.24 dan diagram aktivitas pada Gambar 3.27.

Tabel 3.24 Menggunakan *Reward Event*

Kode	UC-009
Nama	Menggunakan <i>Reward Event</i>
Aktor	<i>Event</i>
Deskripsi	<i>Event</i> dapat menukarkan dan menggunakan <i>reward</i>
Kondisi Awal	<i>Event</i> berada di halaman <i>dashboard event</i>
Kondisi akhir	<i>Event</i> berhasil menggunakan <i>reward</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>Event</i> masuk ke halaman <i>reward</i> pada <i>navbar</i>2. <i>Event</i> mengklik tombol <i>View Reward</i> pada <i>event</i> yang diinginkan3. Sistem menampilkan halaman berisi <i>reward</i> yang tersedia bagi <i>event</i>4. <i>Event</i> menggunakan <i>reward</i> yang diinginkan5. Sistem menampilkan data <i>point user</i> sebelum dan sesudah menukarkan <i>reward</i>6. <i>Event</i> memilih <i>Redeem</i>7. Sistem menampilkan notifikasi berhasil	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">5. Sistem menampilkan data <i>point user</i> tidak cukup5.1 <i>Event</i> memilih <i>Close</i>	



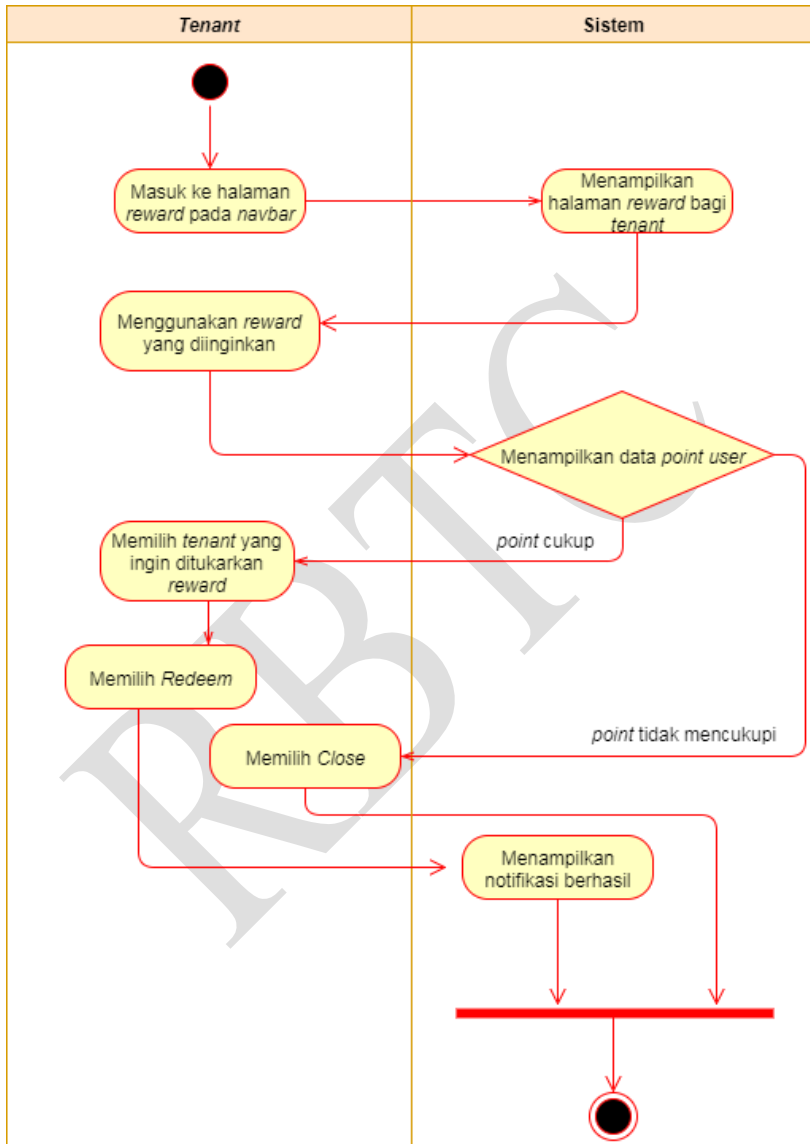
Gambar 3.27 Diagram Aktivitas Menggunakan *Reward Event*

3.2.1.10 Menggunakan *Reward Tenant*

Pada kasus penggunaan ini, *tenant* dapat menukarkan *point*-nya dengan *reward* yang tersedia. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.25 dan diagram aktivitas pada Gambar 3.28.

Tabel 3.25 Menggunakan *Reward Tenant*

Kode	UC-010
Nama	Menggunakan <i>Reward Tenant</i>
Aktor	<i>Tenant</i>
Deskripsi	<i>Tenant</i> dapat menukarkan dan menggunakan <i>reward</i>
Kondisi Awal	<i>Tenant</i> berada di halaman <i>dashboard tenant</i>
Kondisi akhir	<i>Tenant</i> berhasil menggunakan <i>reward</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>Tenant</i> masuk ke halaman <i>Reward</i> pada <i>navbar</i>2. Sistem menampilkan halaman berisi <i>reward</i> yang tersedia bagi <i>tenant</i>3. <i>Tenant</i> mengklik menggunakan <i>reward</i> yang diinginkan4. Sistem menampilkan data <i>point user</i> sebelum dan sesudah menukarkan <i>reward</i>5. <i>Tenant</i> memilih <i>tenant</i> mana yang ingin ditukarkan <i>voucher reward</i>-nya6. <i>Tenant</i> memilih <i>Redeem</i>7. Sistem menampilkan notifikasi berhasil	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">4. Sistem menampilkan data <i>point user</i> tidak cukup4.1 <i>Tenant</i> memilih <i>Close</i>	



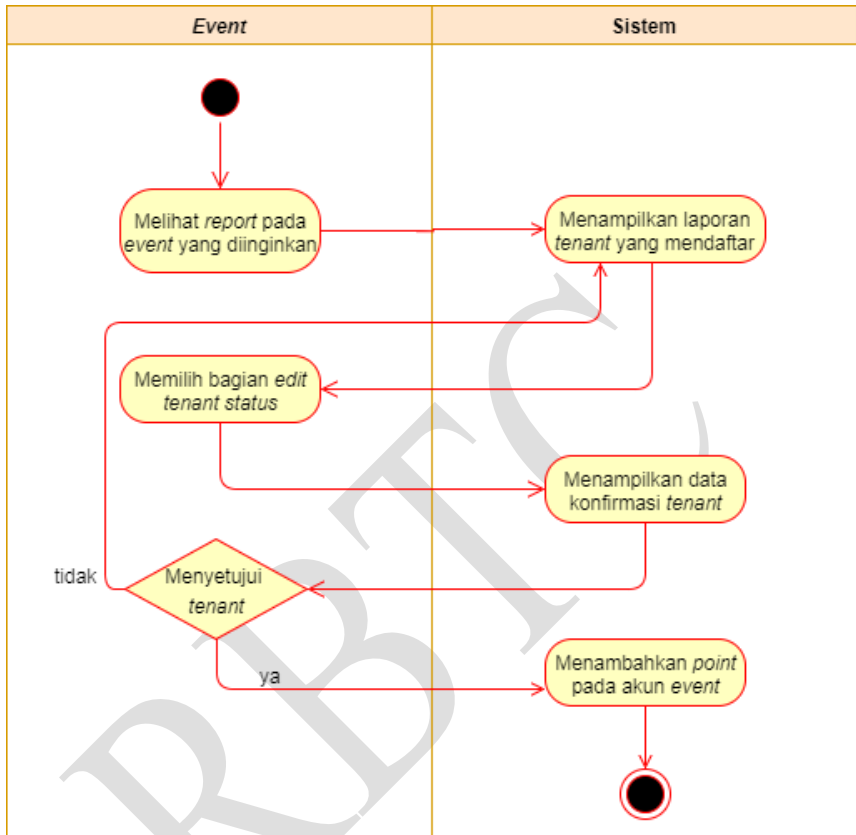
Gambar 3.28 Diagram Aktivitas Menggunakan Reward Tenant

3.2.1.11 Event Mendapatkan Point

Pada kasus penggunaan ini, *event* dapat mendapatkan *point* setelah menyetujui pendaftaran *tenant*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.26 dan diagram aktivitas pada Gambar 3.29.

Tabel 3.26 Event Mendapatkan Point

Kode	UC-011
Nama	<i>Event Mendapatkan Point</i>
Aktor	<i>Event</i>
Deskripsi	<i>Event mendapatkan point setelah menyetujui pendaftaran tenant</i>
Kondisi Awal	<i>Event berada di halaman dashboard event</i>
Kondisi akhir	<i>Event berhasil mendapatkan point</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>Event</i> melihat <i>Report</i> pada <i>event</i> yang diinginkan2. Sistem menampilkan halaman berisi laporan mengenai pendaftaran <i>tenant</i> pada <i>event</i> tersebut3. <i>Event</i> memilih bagian <i>Edit Tenant Status</i>4. Sistem menampilkan data <i>tenant</i> yang belum dan sudah dikonfirmasi5. <i>Event</i> menyetujui <i>tenant</i> yang diinginkan6. Sistem berhasil menambahkan <i>point</i> pada akun <i>Event</i>	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">5. Event tidak menyetujui <i>tenant</i><ol style="list-style-type: none">5.1 Kembali ke nomor 2	



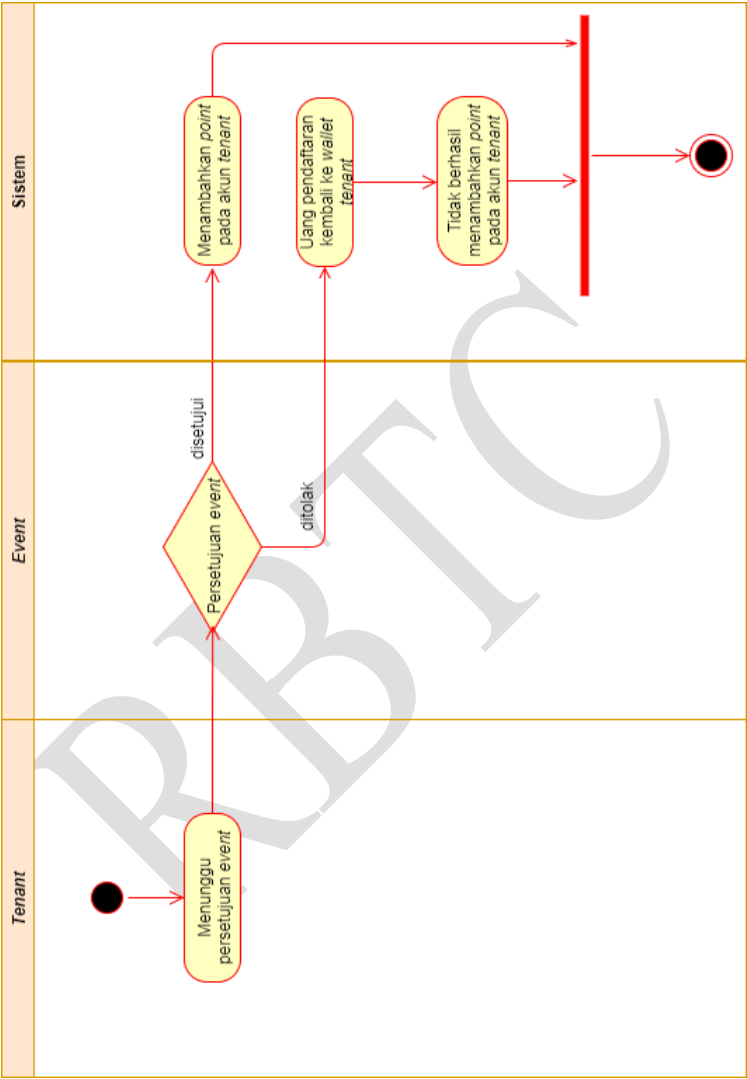
Gambar 3.29 Diagram Aktivitas *Event Mendapatkan Point*

3.2.1.12 *Tenant Mendapatkan Point*

Pada kasus penggunaan ini, *tenant* dapat mendapatkan *point* setelah mendaftar pada *event* dan pendaftarannya disetujui oleh *event* tersebut. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.27 dan diagram aktivitas pada Gambar 3.30.

Tabel 3.27 *Tenant Mendapatkan Point*

Kode	UC-012
Nama	<i>Tenant Mendapatkan Point</i>
Aktor	<i>Tenant</i>
Deskripsi	<i>Tenant</i> mendapatkan <i>point</i> setelah mendaftar dan disetujui pendaftarannya oleh <i>event</i> tersebut
Kondisi Awal	<i>Tenant</i> sudah mendaftarkan diri pada <i>event</i>
Kondisi akhir	<i>Tenant</i> berhasil mendapatkan <i>point</i>
Alur Kejadian Normal	
1. <i>Tenant</i> menunggu persetujuan <i>event</i> 2. <i>Event</i> menyetujui pendaftaran <i>tenant</i> 3. Sistem berhasil menambahkan <i>point</i> pada akun <i>tenant</i>	
Alur Kejadian Alternatif	
2. <i>Event</i> tidak menyetujui pendaftaran <i>tenant</i> 2.1 Uang pendaftaran masuk kembali ke <i>wallet tenant</i> 2.2 Sistem tidak berhasil menambahkan <i>point</i> pada akun <i>tenant</i>	



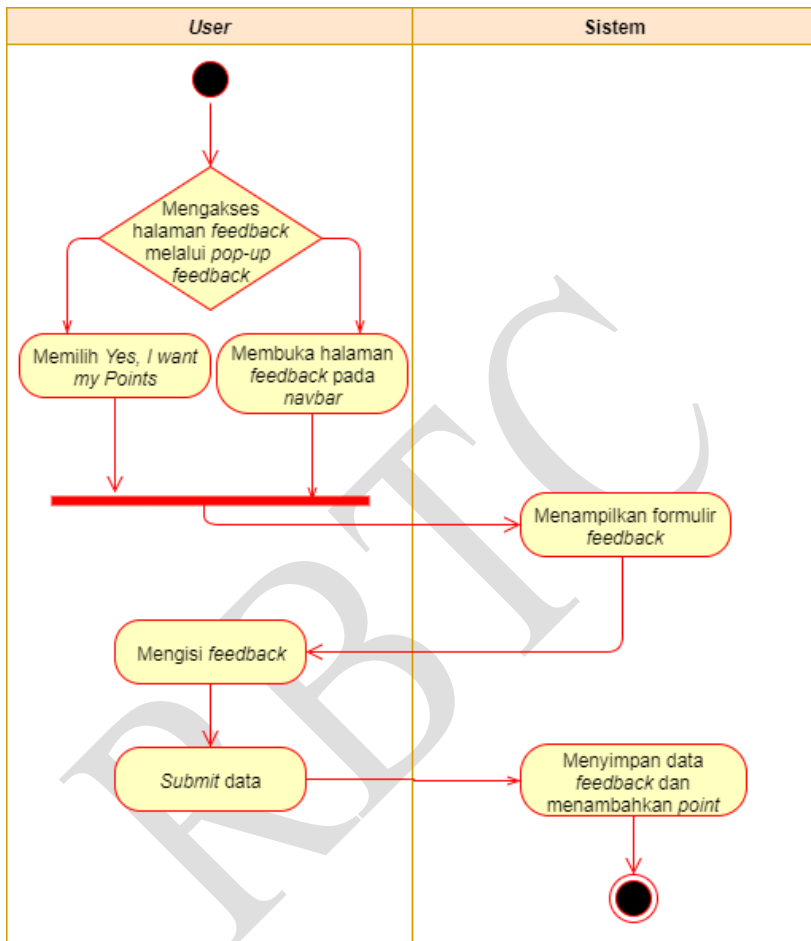
Gambar 3.30 Diagram Aktivitas *Tenant Mendapatkan Point*

3.2.1.13 Mengisi *Feedback*

Pada kasus penggunaan ini, *user* dapat mengisi *feedback* untuk CallTenant. Spesifikasi kasus penggunaan dapat dilihat pada tabel Tabel 3.28 dan diagram aktivitas pada Gambar 3.31.

Tabel 3.28 Mengisi *Feedback*

Kode	UC-013
Nama	Mengisi <i>Feedback</i>
Aktor	<i>User</i>
Deskripsi	<i>User</i> mengisi <i>feedback</i> bagi CallTenant untuk mendapatkan <i>point</i>
Kondisi Awal	<i>User</i> sudah login
Kondisi akhir	Sistem berhasil menyimpan <i>feedback</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>User</i> mengakses halaman <i>feedback</i> melalui <i>pop-up</i>2. <i>User</i> memilih <i>Yes, I want my points</i>3. Sistem menampilkan halaman formulir <i>feedback</i>4. <i>User</i> mengisikan <i>feedback</i>5. <i>User</i> meng-submit data6. Sistem berhasil menyimpan data <i>feedback</i> dan menambahkan <i>point</i>	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">2. <i>User</i> membuka halaman <i>feedback</i> pada navbar2.1 Kembali ke nomor 3	



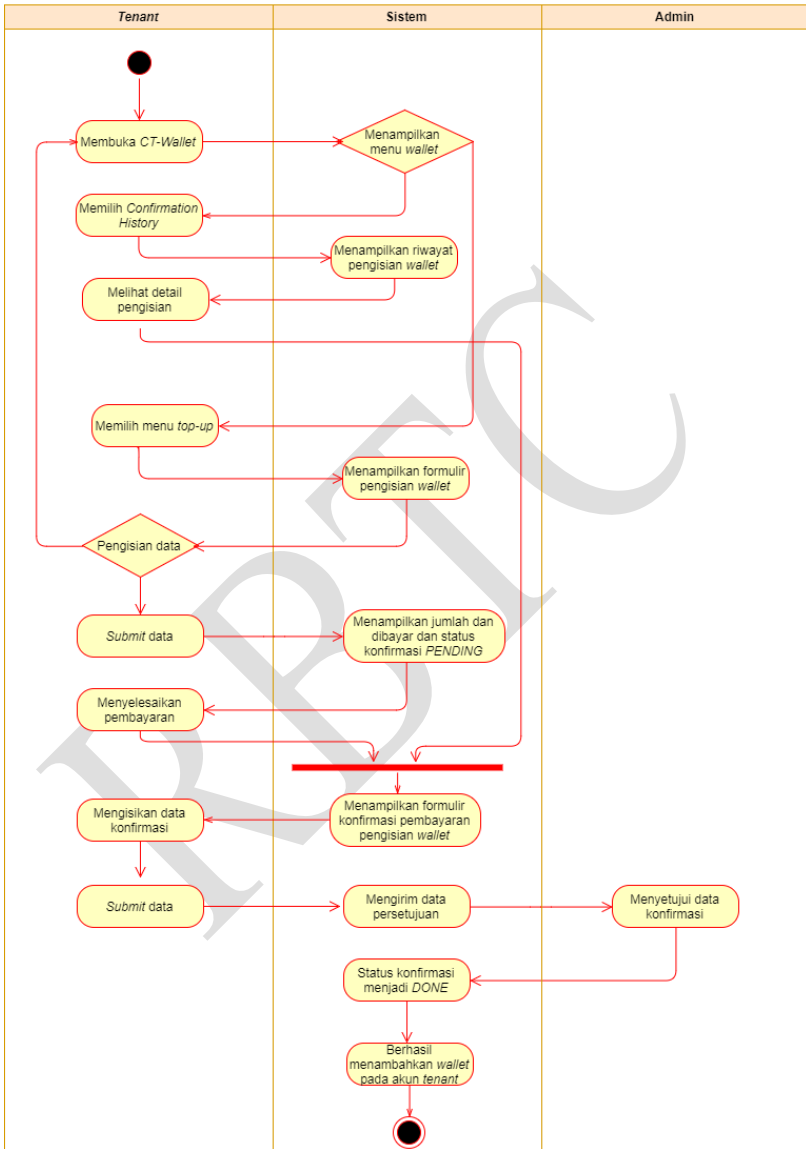
Gambar 3.31 Diagram Aktivitas Mengisi *Feedback*

3.2.1.14 Mengisi *Wallet*

Pada kasus penggunaan ini, *tenant* dapat mengisi *wallet* dengan menyelesaikan pembayaran melalui transfer bank. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.29 dan diagram aktivitas pada Gambar 3.32.

Tabel 3.29 Mengisi Wallet

Kode	UC-014
Nama	Mengisi <i>Wallet</i>
Aktor	<i>Tenant</i>
Deskripsi	<i>Tenant</i> dapat mengisi <i>wallet</i>
Kondisi Awal	<i>Tenant</i> berada pada halaman <i>wallet</i> dan <i>point</i>
Kondisi akhir	<i>Tenant</i> berhasil mengisi <i>wallet</i>
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. <i>Tenant</i> mengklik tombol <i>CT-Wallet</i> 2. Sistem menampilkan menu <i>CT-Wallet</i> 3. <i>Tenant</i> memilih menu <i>top-up</i> 4. Sistem menampilkan formulir pengisian <i>wallet</i> 5. <i>Tenant</i> mengisi data 6. <i>Tenant</i> meng-<i>submit</i> data 7. Sistem menampilkan jumlah yang harus dibayar dan status konfirmasi menjadi <i>PENDING</i> 8. <i>Tenant</i> menyelesaikan pembayaran 9. Sistem menampilkan formulir konfirmasi pembayaran pengisian <i>wallet</i> 10. <i>Tenant</i> mengisi data konfirmasi 11. <i>Tenant</i> meng-<i>submit</i> data 12. Sistem mengirim data persetujuan untuk admin dan status konfirmasi menjadi <i>IN PROGRESS</i> 13. Admin menyetujui data konfirmasi 14. Status konfirmasi <i>wallet</i> berubah menjadi <i>DONE</i> 15. Sistem berhasil menambahkan <i>wallet</i> ke akun <i>tenant</i> 	
Alur Kejadian Alternatif	
<ol style="list-style-type: none"> 10. <i>Tenant</i> tidak mengisi data konfirmasi <ol style="list-style-type: none"> 10.1 Kembali ke nomor 1 10.2 Sistem menampilkan menu <i>wallet</i> 10.3 <i>Tenant</i> mengklik tombol <i>Confirmation History</i> 10.4 Sistem menampilkan riwayat pengisian <i>wallet</i> 10.5 <i>Tenant</i> melihat detail pada data yang berstatus <i>PENDING</i> 10.6 Kembali ke nomor 9 	



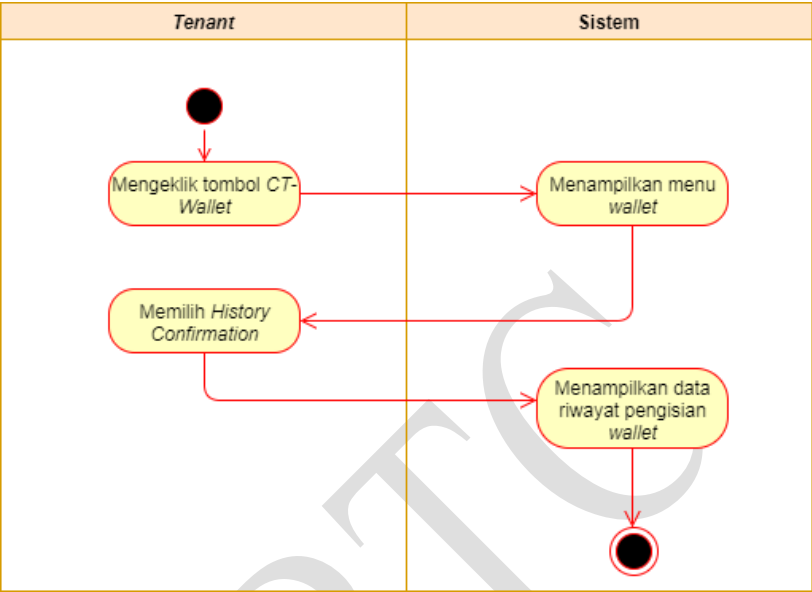
Gambar 3.32 Diagram Aktivitas Mengisi Wallet

3.2.1.15 Melihat Riwayat Konfirmasi *Top-up Wallet*

Pada kasus penggunaan ini, *tenant* dapat mengisi *wallet* dengan menyelesaikan pembayaran melalui transfer bank. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.30 dan diagram aktivitas pada Gambar 3.33.

Tabel 3.30 Melihat Riwayat Konfirmasi *Top-up Wallet*

Kode	UC-015
Nama	Melihat Riwayat Konfirmasi <i>Top-up Wallet</i>
Aktor	<i>Tenant</i>
Deskripsi	<i>Tenant</i> dapat melihat riwayat konfirmasi pengisian <i>wallet</i>
Kondisi Awal	<i>Tenant</i> berada pada halaman <i>wallet</i>
Kondisi akhir	<i>Tenant</i> berhasil melihat riwayat konfirmasi pengisian <i>wallet</i>
Alur Kejadian Normal	
1. <i>Tenant</i> mengeklik tombol <i>CT-Wallet</i> 2. Sistem menampilkan menu <i>wallet</i> 3. <i>Tenant</i> mengeklik tombol <i>History Confirmation</i> 4. Sistem menampilkan data riwayat pengisian <i>wallet</i>	
Alur Kejadian Alternatif	
-	



Gambar 3.33 Diagram Aktivitas Melihat Riwayat Konfirmasi Top-up Wallet

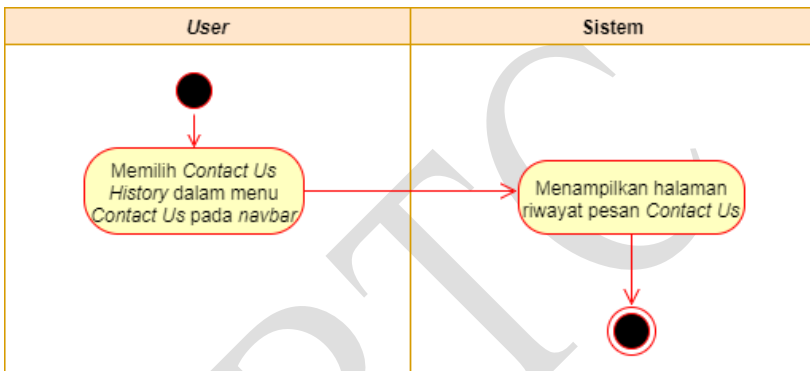
3.2.1.16 Melihat Riwayat Pesan Contact Us

Pada kasus penggunaan ini, *user* baik *tenant* maupun *event* dapat melihat riwayat pesan dan balasannya yang pernah dikirim melalui fitur *Contact Us*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.31 dan aktivitas diagram pada Gambar 3.34.

Tabel 3.31 Melihat Riwayat Pesan Contact Us

Kode	UC-016
Nama	Melihat Riwayat Pesan pada <i>Contact Us</i>
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melihat riwayat pesan dan balasannya yang pernah dikirim melalui <i>Contact Us</i>
Kondisi Awal	<i>User</i> sudah <i>login</i>
Kondisi akhir	<i>User</i> berhasil melihat riwayat pesan <i>Contact Us</i>

Alur Kejadian Normal
1. <i>User</i> memilih <i>Contact Us History</i> dalam menu <i>Contact Us</i> pada <i>navbar</i>
2. Sistem menampilkan halaman data riwayat pesan <i>contact us</i>
Alur Kejadian Alternatif
-



Gambar 3.34 Diagram Aktivitas Melihat Riwayat Pesan *Contact Us*

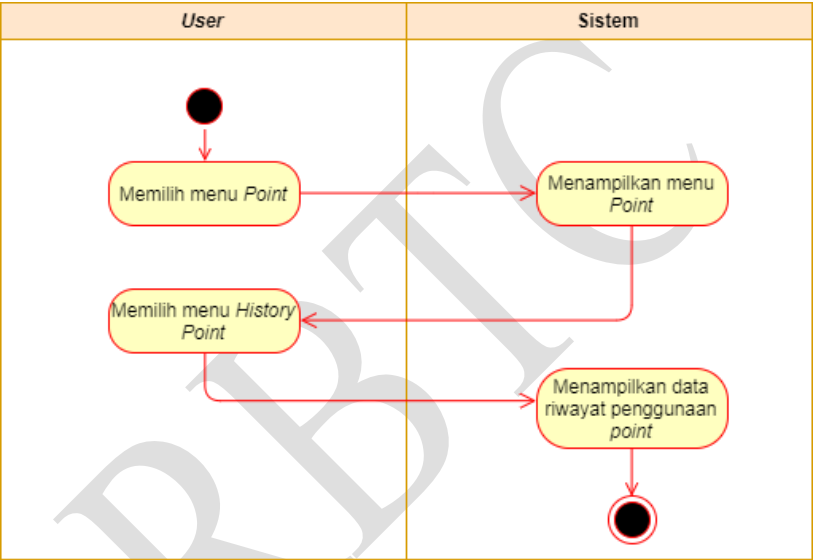
3.2.1.17 Melihat Riwayat Pemakaian *Point*

Pada kasus penggunaan ini, *user* baik *tenant* maupun *event* dapat melihat riwayat penggunaan *point*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.32 dan diagram aktivitas pada Gambar 3.35.

Tabel 3.32 Melihat Riwayat Pemakaian *Point*

Kode	UC-017
Nama	Melihat Riwayat Pemakaian <i>Point</i>
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melihat riwayat penggunaan <i>point</i>
Kondisi Awal	<i>User</i> berada pada halaman <i>wallet</i> dan <i>point</i>
Kondisi akhir	<i>User</i> berhasil melihat riwayat penggunaan <i>point</i>

Alur Kejadian Normal
1. User memilih menu <i>Point</i>
2. Sistem menampilkan menu <i>point</i>
3. User memilih menu <i>History Point</i>
4. Sistem menampilkan data riwayat penggunaan <i>point</i>
Alur Kejadian Alternatif
-



Gambar 3.35 Diagram Aktivitas Melihat Riwayat Pemakaian *Point*

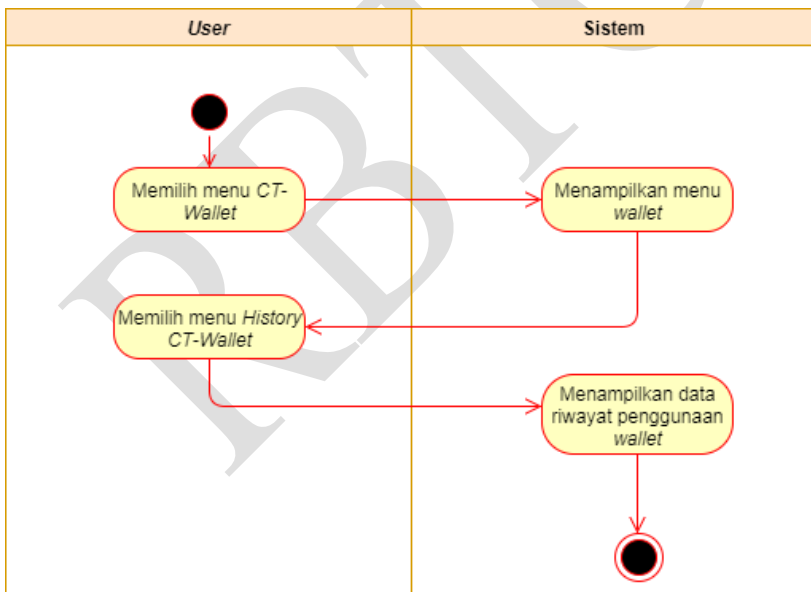
3.2.1.18 Melihat Riwayat Pemakaian *Wallet*

Pada kasus penggunaan ini, *user* baik *tenant* maupun *event* dapat melihat riwayat penggunaan *point*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.33 dan diagram aktivitas pada Gambar 3.36.

Tabel 3.33 Melihat Riwayat Pemakaian *Wallet*

Kode	UC-018
Nama	Melihat Riwayat Pemakaian <i>Wallet</i>

Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melihat riwayat penggunaan <i>wallet</i>
Kondisi Awal	<i>User</i> berada pada halaman <i>wallet</i> dan <i>point</i>
Kondisi akhir	<i>User</i> berhasil melihat riwayat penggunaan <i>wallet</i>
Alur Kejadian Normal	
1. <i>User</i> memilih menu <i>CT-Wallet</i> 2. Sistem menampilkan menu <i>wallet</i> 3. <i>User</i> memilih menu <i>History CT-Wallet</i> 4. Sistem menampilkan data riwayat penggunaan <i>wallet</i>	
Alur Kejadian Alternatif	
-	



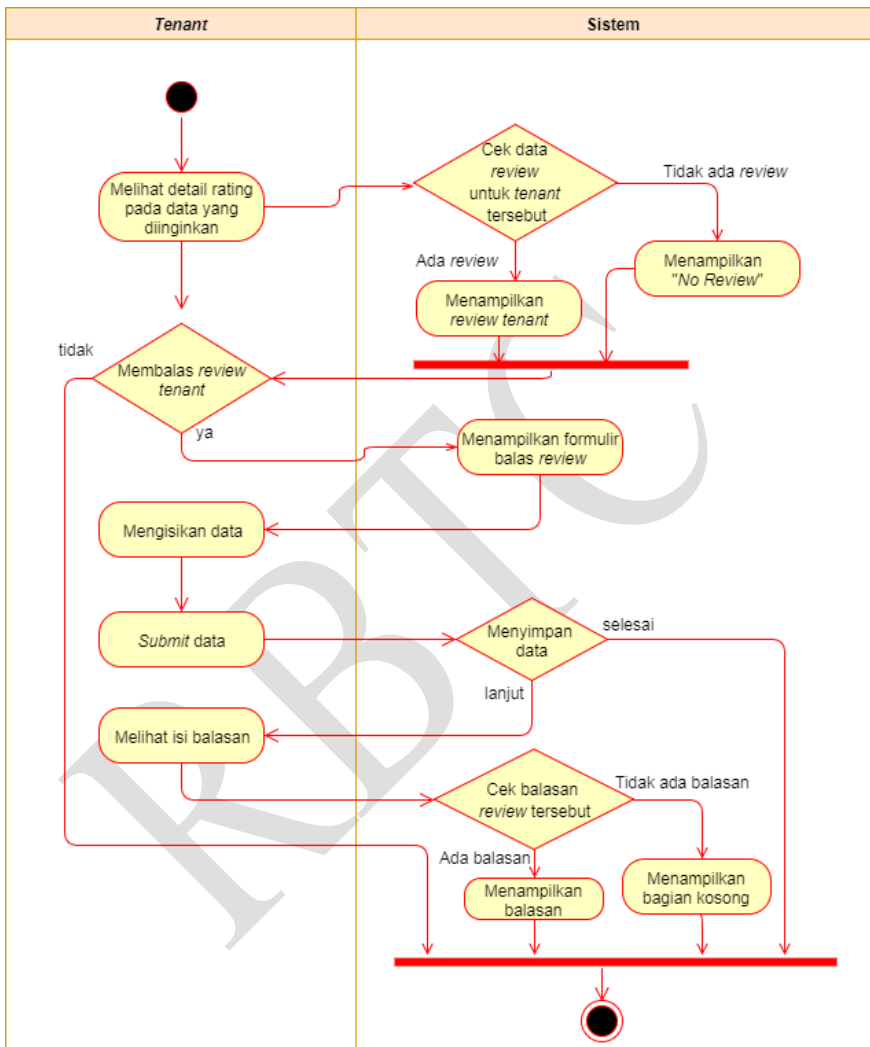
Gambar 3.36 Diagram Aktivitas Melihat Riwayat Pemakaian Wallet

3.2.1.19 Mengelola Data Review Tenant

Pada kasus penggunaan ini, mengelola data *review tenant* dapat terdiri dari melihat semua *review* yang tertuju pada *tenant* tersebut serta membalas *review*-nya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.34 dan diagram aktivitas pada Gambar 3.37.

Tabel 3.34 Mengelola Data Review Tenant

Kode	UC-019
Nama	Mengelola Data Review Tenant
Aktor	Tenant
Deskripsi	Tenant dapat melihat semua <i>review</i> yang tertuju pada <i>tenant</i> tersebut serta membalas <i>review</i> tersebut
Kondisi Awal	Tenant berada pada profil <i>tenant</i>
Kondisi akhir	Tenant berhasil mengelola data <i>review tenant</i>
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Tenant melihat detail <i>rating</i> pada <i>tenant</i> yang diinginkan 2. Sistem menampilkan semua data <i>review</i> untuk <i>tenant</i> tersebut 3. Tenant membalas <i>review</i> 4. Sistem menampilkan formulir balas pesan 5. Tenant mengisikan data 6. Tenant meng-submit data 7. Sistem berhasil menyimpan data 8. Tenant melihat isi balasan 9. Sistem mengecek balasan <i>review</i> tersebut 	
Alur Kejadian Alternatif	
<ol style="list-style-type: none"> 2. Sistem mendeteksi tidak ada data <i>review</i> untuk <i>tenant</i> tersebut <ol style="list-style-type: none"> 2.1 Sistem menampilkan keterangan “No Review” 9. Sistem menampilkan tidak ada balasan <ol style="list-style-type: none"> 9.1 Sistem menampilkan bagian kosong 	



Gambar 3.37 Diagram Aktivitas Mengelola Data *Review Tenant*

3.2.1.20 Mengelola Data Review Event

Pada kasus penggunaan ini, mengelola data *review event* terdiri dari melihat semua *review* yang tertuju pada *event* tersebut serta membalas *review*-nya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.35 dan diagram aktivitas pada Gambar 3.38.

Tabel 3.35 Mengelola Data Review Event

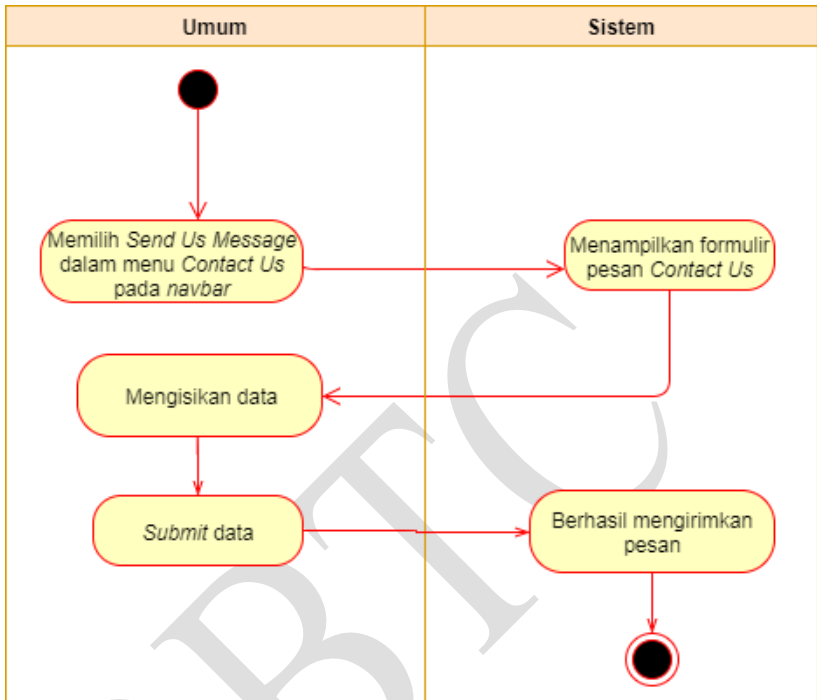
Kode	UC-020
Nama	Mengelola Data Review Event
Aktor	Event
Deskripsi	Event dapat melihat semua review yang tertuju pada event tersebut serta membalas
Kondisi Awal	Event berada pada dashboard event
Kondisi akhir	Event berhasil mengelola data review event
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Event melihat detail <i>rating</i> pada event yang diinginkan 2. Sistem menampilkan semua data <i>review</i> untuk event tersebut 3. Event membalas <i>review</i> 4. Sistem menampilkan formulir balas pesan 5. Event mengisikan data 6. Event meng-submit data 7. Sistem berhasil menyimpan data 8. Event melihat balasan <i>review</i> 9. Sistem mengecek balasan <i>review</i> tersebut 	
Alur Kejadian Alternatif	
<ol style="list-style-type: none"> 2. Sistem mendeteksi tidak ada data <i>review</i> untuk event tersebut <ol style="list-style-type: none"> 2.1 Sistem menampilkan keterangan “No Review” 9. Sistem menampilkan tidak ada balasan <ol style="list-style-type: none"> 9.1 Sistem menampilkan bagian kosong 	

3.2.1.21 Menggunakan Fitur *Contact Us*

Pada kasus penggunaan ini, *user* baik *non-member*, *tenant* maupun *event* dapat mengirimkan pesan melalui fitur *Contact Us*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.36 dan diagram aktivitas pada Gambar 3.39.

Tabel 3.36 Menggunakan Fitur *Contact Us*

Kode	UC-021
Nama	Menggunakan Fitur <i>Contact Us</i>
Aktor	Umum
Deskripsi	Pengguna dapat mengirimkan pesan melalui fitur <i>Contact Us</i>
Kondisi Awal	Pengguna berada pada web CallTenant
Kondisi akhir	Pengguna berhasil mengirim pesan melalui <i>Contact Us</i>
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna umum memilih <i>Send Us Message</i> dalam menu <i>Contact Us</i> pada <i>navbar</i> 2. Sistem menampilkan formulir pesan <i>Contact Us</i> 3. Pengguna mengisikan data 4. Pengguna meng-<i>submit</i> data 5. Sistem berhasil mengirimkan pesan 	
Alur Kejadian Alternatif	
-	



Gambar 3.39 Diagram Aktivitas Menggunakan Fitur *Contact Us*

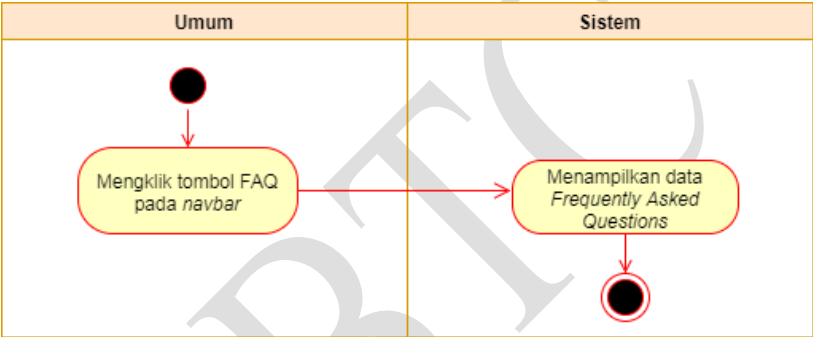
3.2.1.22 Menggunakan Fitur *Frequently Asked Questions*

Pada kasus penggunaan ini, *user* baik *non-member*, *tenant* maupun *event* dapat melihat data *Frequently Asked Questions*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.37 dan diagram aktivitas pada Gambar 3.40.

Tabel 3.37 Menggunakan Fitur *Frequently Asked Questions*

Kode	UC-022
Nama	Menggunakan Fitur <i>Frequently Asked Questions</i>
Aktor	Umum
Deskripsi	Pengguna dapat melihat data <i>Frequently Asked Questions</i>

Kondisi Awal	Pengguna berada pada web CallTenant
Kondisi akhir	Pengguna berhasil mengirim pesan melalui <i>Frequently Asked Questions</i>
Alur Kejadian Normal	
1. Pengguna umum mengklik tombol FAQ pada <i>navbar</i> 2. Sistem menampilkan data <i>Frequently Asked Questions</i>	
Alur Kejadian Alternatif	
-	



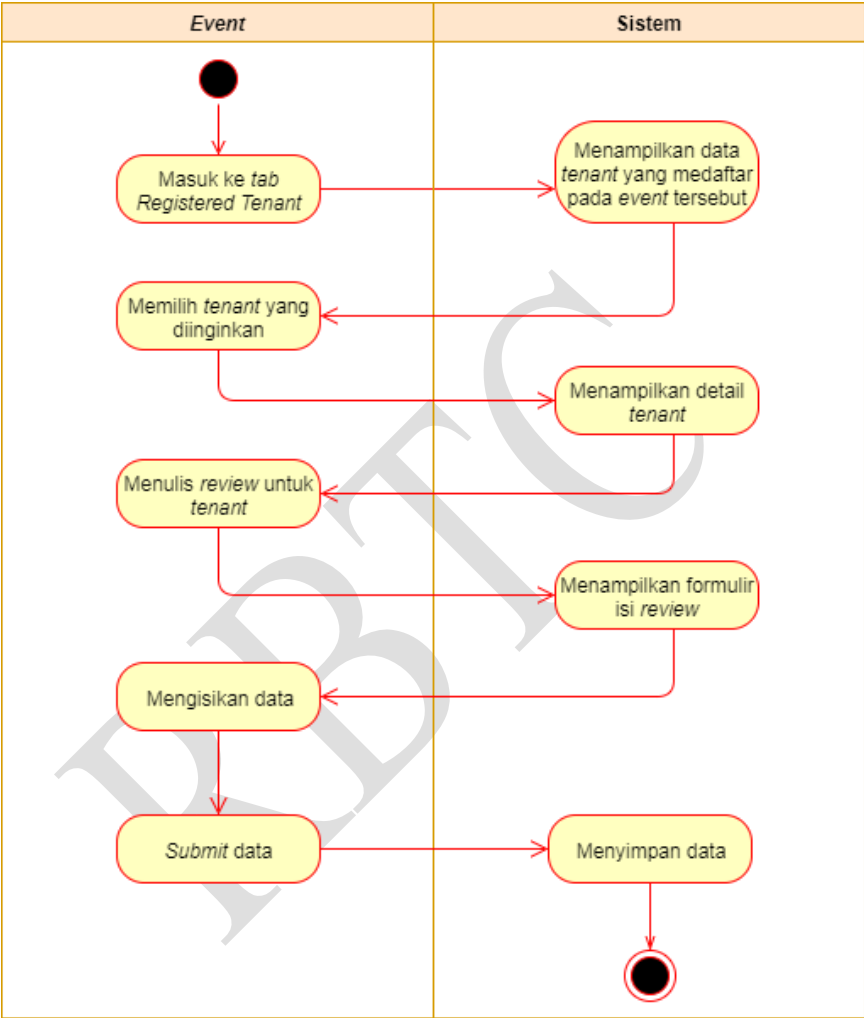
Gambar 3.40 Diagram Aktivitas Menggunakan Fitur *Frequently Asked Questions*

3.2.1.23 Mengisi *Review* untuk *Tenant*

Pada kasus penggunaan ini, *event* dapat mengisi *review* untuk *tenant* yang pernah mendaftar padanya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.38 dan diagram aktivitas pada Gambar 3.41.

Tabel 3.38 Mengisi *Review* untuk *Tenant*

Kode	UC-023
Nama	Mengisi <i>Review</i> untuk <i>Tenant</i>
Aktor	<i>Event</i>
Deskripsi	<i>Event</i> dapat mengisi <i>review</i> untuk <i>tenant</i> yang terdaftar padanya
Kondisi Awal	<i>Event</i> berada pada halaman <i>report</i>
Kondisi akhir	<i>Event</i> berhasil mengisi <i>review</i> untuk <i>tenant</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>Event</i> masuk ke tab <i>Registered Tenant</i>2. Sistem menampilkan data <i>tenant</i> yang mendaftar pada <i>event</i> tersebut3. <i>Event</i> memilih <i>tenant</i> yang diinginkan4. Sistem menampilkan detail <i>tenant</i>5. <i>Event</i> menulis <i>review</i> untuk <i>tenant</i>6. Sistem menampilkan formulir isi <i>review</i>7. <i>Event</i> mengisikan data8. <i>Event</i> meng-submit data9. Sistem menyimpan data	
Alur Kejadian Alternatif	
-	



Gambar 3.41 Diagram Aktivitas Mengisi *Review* untuk *Tenant*

3.2.1.24 Mengisi *Review* untuk *Event*

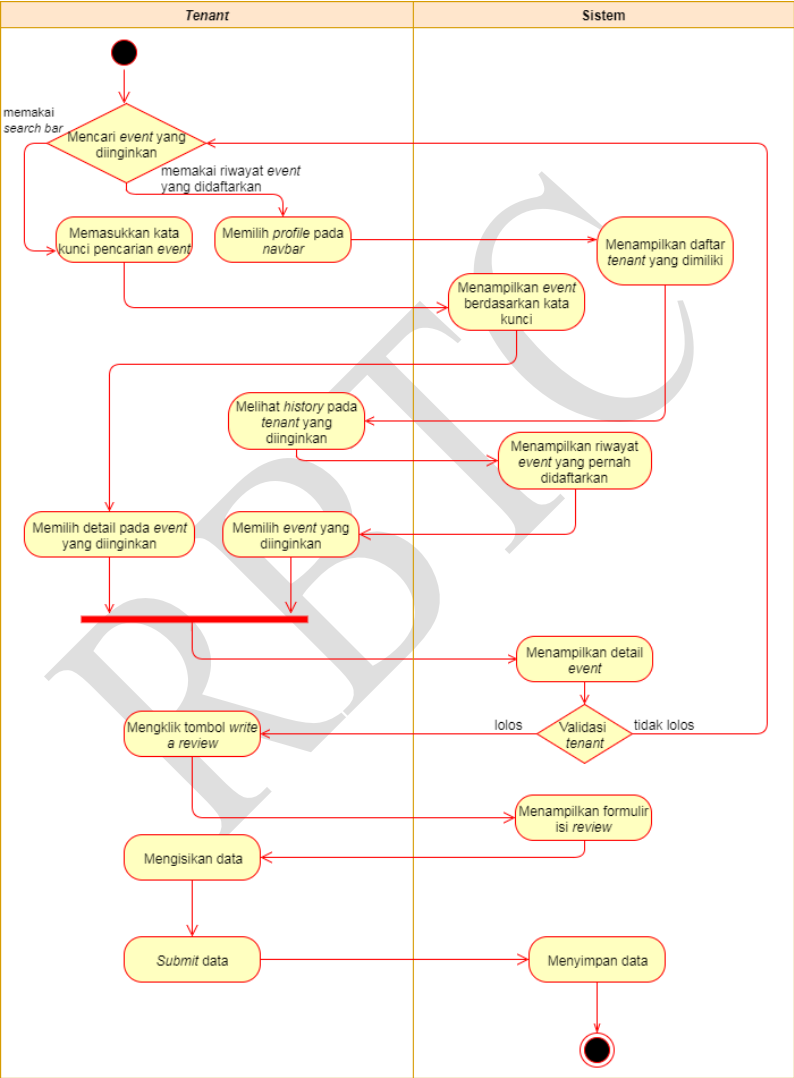
Pada kasus penggunaan ini, *event* dapat mengisi *review* untuk *tenant* yang pernah mendaftar padanya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.39 dan diagram aktivitas pada Gambar 3.42.

Tabel 3.39 Mengisi *Review* untuk *Event*

Kode	UC-024
Nama	Mengisi <i>Review</i> untuk <i>Event</i>
Aktor	<i>Tenant</i>
Deskripsi	<i>Tenant</i> dapat mengisi <i>review</i> untuk <i>event</i> yang pernah didaftarkan
Kondisi Awal	<i>Tenant</i> berada dalam <i>dashboard tenant</i>
Kondisi akhir	<i>Tenant</i> berhasil mengisi <i>review</i> untuk <i>event</i>
Alur Kejadian Normal	
<ol style="list-style-type: none">1. <i>Tenant</i> mencari <i>event</i> yang diinginkan2. <i>Tenant</i> memasukkan kata kunci pencarian <i>event</i>3. Sistem menampilkan <i>event</i> berdasarkan kata kunci4. <i>Tenant</i> mengeklik detail pada <i>event</i> yang diinginkan5. Sistem menampilkan detail <i>event</i>6. Sistem lolos validasi pendaftaran <i>tenant</i> pada <i>event</i>7. <i>Tenant</i> menulis <i>review</i> untuk <i>event</i>8. Sistem menampilkan formulir isi <i>review</i>9. <i>Tenant</i> mengisi data10. <i>Tenant</i> meng-submit data11. Sistem menyimpan data	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">1. <i>Tenant</i> melihat <i>event</i> yang sudah pernah didaftarkan<ol style="list-style-type: none">1.1 <i>Tenant</i> memilih <i>Profile</i> dalam menu <i>welcome</i> pada <i>navbar</i>1.2 Sistem menampilkan daftar <i>tenant</i> yang dimiliki1.3 <i>Tenant</i> memilih <i>history</i> pada <i>tenant</i> yang diinginkan1.4 Sistem menampilkan riwayat <i>event</i> yang pernah didaftarkan1.5 <i>Tenant</i> memilih <i>event</i> yang diinginkan1.6 Kembali ke nomor 4	

6. Sistem tidak lolos validasi

6.1 Kembali ke nomor 1



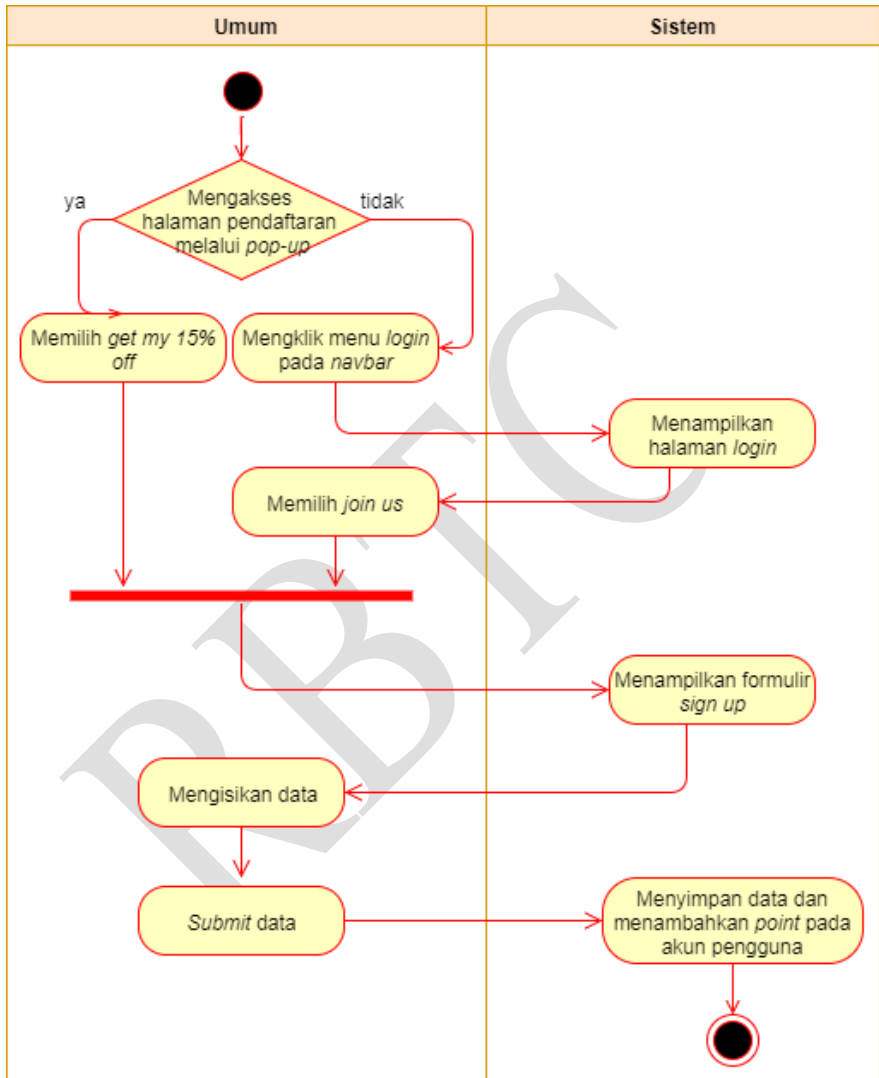
Gambar 3.42 Diagram Aktivitas Mengisi *Review* untuk *Event*

3.2.1.25 Mendaftar sebagai *Event* atau *Tenant*

Pada kasus penggunaan ini, pengguna dapat mendaftar sebagai *event* atau *tenant* pada CallTenant. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.40 dan diagram aktivitas pada Gambar 3.43.

Tabel 3.40 Mendaftar sebagai *Event* atau *Tenant*

Kode	UC-025
Nama	Mendaftar sebagai <i>Event</i> atau <i>Tenant</i>
Aktor	Umum
Deskripsi	Pengguna umum dapat mendaftarkan diri pada CallTenant dan mendapatkan <i>point</i> sebagai bonus pendaftaran pertama
Kondisi Awal	Pengguna sudah berada pada halaman utama CallTenant
Kondisi akhir	Pengguna berhasil mendaftar
Alur Kejadian Normal	
<ol style="list-style-type: none">1. Pengguna mengakses halaman pendaftaran melalui <i>pop-up</i>2. Pengguna memilih <i>get my 15% off</i>3. Sistem menampilkan formulir <i>sign up</i>4. Pengguna mengisi data5. Pengguna mengklik tombol <i>register</i>6. Sistem menyimpan data dan menambahkan <i>point</i> pada akun pengguna	
Alur Kejadian Alternatif	
<ol style="list-style-type: none">2. Pengguna memilih menu <i>login</i> pada <i>navbar</i><ol style="list-style-type: none">2.1 Pengguna mengklik menu <i>login</i> pada <i>navbar</i>2.2 Sistem menampilkan halaman <i>login</i>2.3 Pengguna mengklik tombol <i>Join Us</i>2.4 Kembali ke nomor 3	



Gambar 3.43 Diagram Aktivitas Mendaftar sebagai *Event* atau *Tenant*

3.2.2 Perancangan Basis Data

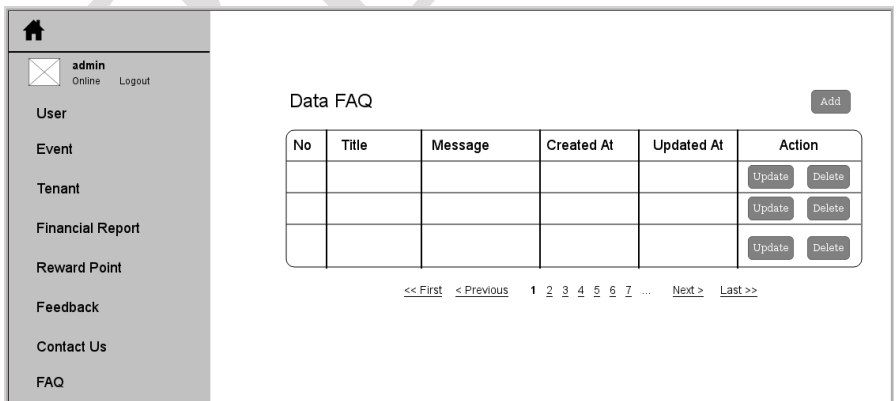
Pada subbab ini dijelaskan mengenai perancangan basis data yang dalam hal ini digunakan untuk menyimpan data diri pengguna baik admin, *event* maupun *tenant*. Dijelaskan juga mengenai proses transaksi antara *event* dan *tenant* serta konfirmasi dari admin, riwayat *review event* dan *tenant*, riwayat *point event* dan *tenant*, riwayat pemakaian *reward* oleh *event* maupun *tenant*, riwayat transaksi yang dilakukan *event* maupun *tenant*, serta pengaturan perhitungan *point*. Gambaran perancangan basis data selengkapnya dapat dilihat pada Lampiran 5. Tabel yang berwarna hijau menandakan bahwa tabel tersebut juga digunakan dalam pembuatan modul *form* dinamis.

3.2.3 Perancangan Antarmuka

Subbab ini menjelaskan bagaimana rancangan antarmuka berinteraksi langsung dengan pengguna.

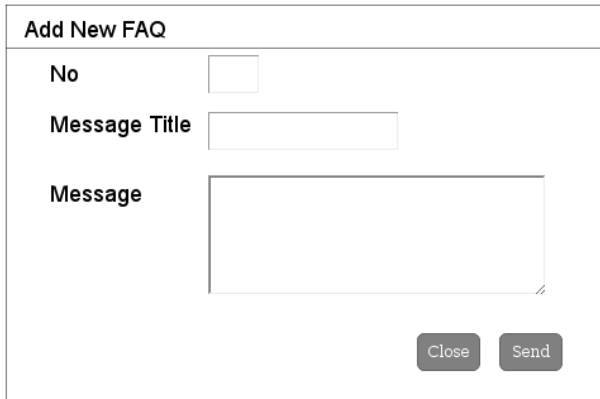
3.2.3.1 Rancangan Antarmuka Mengelola Data *Frequently Asked Questions*

Gambar 3.44 menunjukkan data *Frequently Asked Questions* dan terdapat tiga tombol yaitu *add* pada kanan atas, *Update*, serta *Delete* pada tabel data FAQ.



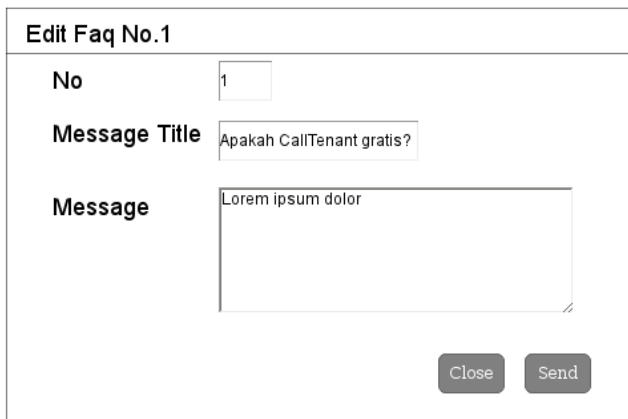
Gambar 3.44 Rancangan Antarmuka Halaman *Frequently Asked Questions*

Formulir menambah pada Gambar 3.45 dan mengubah data *Frequently Asked Questions* pada Gambar 3.46 merupakan formulir pengisian data serta dilengkapi dua tombol untuk *submit* yaitu *Send* dan *Close* untuk kembali ke data *Frequently Asked Questions*.



The image shows a web form titled "Add New FAQ". It contains three input fields: "No" (a small text box), "Message Title" (a medium text box), and "Message" (a large text area). At the bottom right, there are two buttons labeled "Close" and "Send".

Gambar 3.45 Rancangan Antarmuka Menambah Data *Frequently Asked Questions*



The image shows a web form titled "Edit Faq No.1". It contains three input fields: "No" (a small text box with the value "1"), "Message Title" (a medium text box with the value "Apakah CallTenant gratis?"), and "Message" (a large text area with the value "Lorem ipsum dolor"). At the bottom right, there are two buttons labeled "Close" and "Send".

Gambar 3.46 Rancangan Antarmuka Mengubah Data *Frequently Asked Questions*

3.2.3.2 Rancangan Antarmuka Mengelola Pelaporan Keuangan

Halaman laporan keuangan secara keseluruhan terdapat grafik data keuangan dalam satu tahun serta target yang sudah dicapai yang digambarkan dengan *progress bar* seperti pada Gambar 3.47.



Gambar 3.47 Rancangan Antarmuka Laporan Keseluruhan Laporan Keuangan

Halaman data transaksi *user* merupakan data-data *user* yang pernah melakukan transaksi pendaftaran antar *event* dan *tenant*. Terdapat tombol Detail di setiap datanya seperti yang digambarkan pada Gambar 3.48.

Data Payment

Payment for Event	Date	Total Payment	Random Number	5%	Detail
					Detail

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.48 Rancangan Antarmuka Data Transaksi User

Detail dari setiap data transaksi *user* digambarkan seperti pada Gambar 3.49.

Detail Payment

Payment from tenant A to event A

Payment Amount	142351
Unique Code	142351
5%	142351
Total Payment	142351
Net Profit	142351

Close

Gambar 3.49 Rancangan Antarmuka Detail Data Transaksi *User*

Halaman data *wallet* merupakan data-data *user* yang pernah melakukan transaksi dengan *wallet*. Terdapat satu tombol Detail untuk melihat detail transaksi tersebut seperti yang digambarkan pada Gambar 3.50.

admin

Online Logout

User

Event

Tenant

Financial Report

Reward Point

Feedback

Contact Us

FAQ

Data Wallet

Username	Wallet	Detail
		Detail

<< First

< Previous

1

2

3

4

5

6

7

...

Next >

Last >>

Gambar 3.50 Rancangan Antarmuka Halaman Data *Wallet User*

Detail dari data *wallet user* digambarkan seperti pada Gambar 3.51.

Detail Wallet

Date	Amount	Type	Detail

CloseSend

Gambar 3.51 Rancangan Antarmuka Detail Data *Wallet User*

Halaman konfirmasi pembayaran transaksi *user* merupakan data transaksi yang perlu dikonfirmasi oleh admin setelah *tenant* melakukan pembayaran pendaftaran. Pada data dengan status *pending* terdapat tiga tombol yaitu *Accept*, *Delete*, serta *Detail* untuk melihat detail konfirmasi. Sedangkan, pada data dengan status selain *pending* hanya terdapat satu tombol *Detail* seperti yang digambarkan pada Gambar 3.52.

admin
Online Logout

User

Event

Tenant

Financial Report

Reward Point

Feedback

Contact Us

FAQ

Confirmation Payment

ID Pembayaran	Payment Date	Tenant Name	Amount	Status	Action	Detail
					AcceptReject	Detail
						Detail
						Detail

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.52 Rancangan Antarmuka Konfirmasi Pembayaran Transaksi *User*

Halaman konfirmasi pembayaran pengisian *wallet* merupakan data transaksi yang perlu dikonfirmasi oleh admin setelah *tenant* melakukan pembayaran pengisian *wallet*. Pada data dengan status *in progress* terdapat tiga tombol yaitu *Accept*, *Delete*, serta *Detail* untuk melihat detail konfirmasi sedangkan pada data dengan status selain *in progress* hanya terdapat satu tombol *Detail* seperti yang dijelaskan pada Gambar 3.53.

ID Confirmation	Username	Payment Date	Amount	Status	Action	Detail
				PENDING	Accept Reject	Detail
				ACCEPTED		Detail
				REJECTED		Detail

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.53 Rancangan Antarmuka Konfirmasi *Top-up Wallet*

Halaman detail konfirmasi pembayaran *wallet* maupun transaksi antar *tenant* dan *event* digambarkan seperti pada Gambar 3.54.

Payment Confirmation Tenant : A

Payment Date 15/05/2017

Bank Origin BCA

Bank Destination Mandiri

Amount 5345233

Receipt

[Close](#)

Gambar 3.54 Rancangan Antarmuka Detail Konfirmasi Pembayaran Transaksi *User* dan Konfirmasi Pembayaran *Top-up Wallet*

3.2.3.3 Rancangan Antarmuka Membalas Pesan *Contact Us*

Halaman membalas pesan *Contact Us* pada Gambar 3.55 merupakan data pesan yang telah masuk ke dalam CallTenant. Terdapat dua tombol *Reply* untuk mengakses formulir balas pesan yang memiliki status *NOT REPLIED* dan tombol Detail pada data yang memiliki status *REPLIED*.

Message Title	Username	Sender Email	Date/Time	Status	Action
				NOT REPLIED	<button>Reply</button>
				REPLIED	<button>Detail</button>
				REPLIED	<button>Detail</button>

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.55 Rancangan Antarmuka Halaman Data *Contact Us*

Formulir balas pesan terdiri dari dua tombol yaitu, *Send* untuk *submit* formulir dan *Close* untuk kembali ke data *Contact Us* seperti pada Gambar 3.56.

Response Contact Us

Email
joreng@gmail.com

Message Title
Apakah CallTenant gratis?

Message
Bisakah saya daftar tanpa membayar?

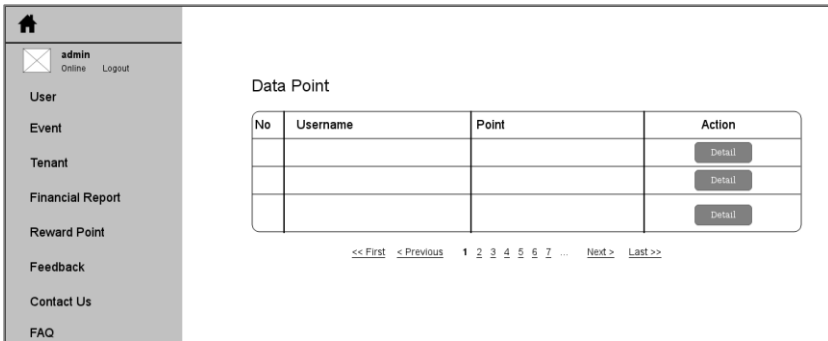
Response

Close Send

Gambar 3.56 Rancangan Antarmuka Formulir Balas Pesan *Contact Us*

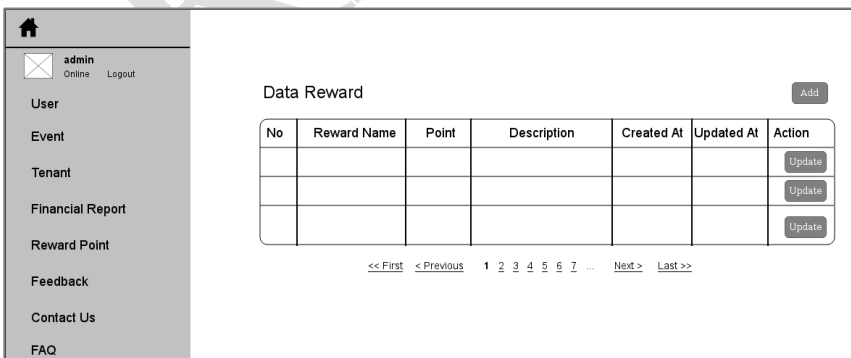
3.2.3.4 Rancangan Antarmuka Mengelola Data *Reward* dan *Point*

Halaman data *point* merupakan data-data *user* yang pernah melakukan transaksi dengan *point*. Terdapat satu tombol Detail di setiap datanya untuk melihat detail penggunaan tersebut seperti pada Gambar 3.57.



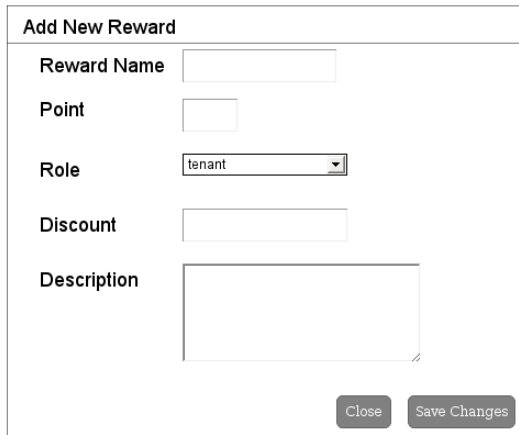
Gambar 3.57 Rancangan Antarmuka Data *Point* User

Halaman data *reward* merupakan data *reward* yang terdaftar dalam CallTenant. Terdapat dua tombol yaitu, *Add* untuk menambah dan *Update* untuk mengubah di setiap datanya yang digambarkan pada Gambar 3.58.



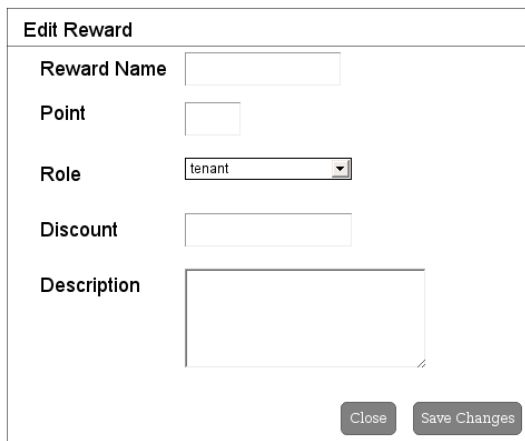
Gambar 3.58 Rancangan Antarmuka Data *Reward*

Formulir menambah data *reward* pada Gambar 3.59 dan formulir mengubah data *reward* pada Gambar 3.60 merupakan formulir pengisian data untuk *event* atau *tenant* sesuai yang dipilih dalam bentuk *dropdown* serta dilengkapi dua tombol untuk submit yaitu *Save Changes* dan *Close* untuk kembali ke data *Reward*.



The form is titled "Add New Reward". It contains five input fields: "Reward Name" (text), "Point" (text), "Role" (dropdown menu with "tenant" selected), "Discount" (text), and "Description" (text area). At the bottom right, there are two buttons: "Close" and "Save Changes".

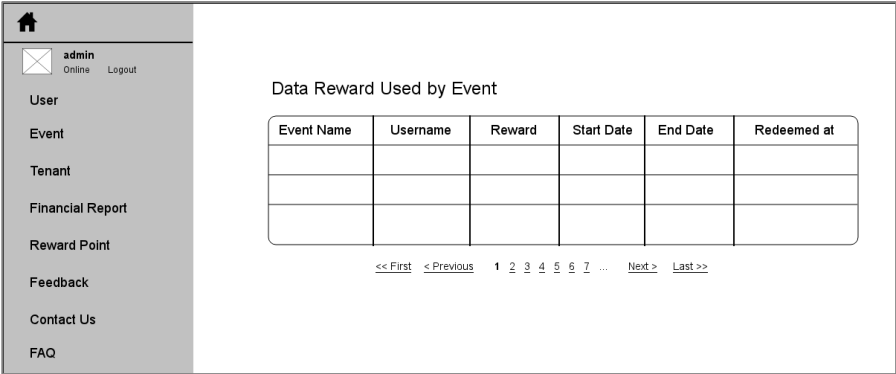
Gambar 3.59 Rancangan Antarmuka Menambah Data *Reward*



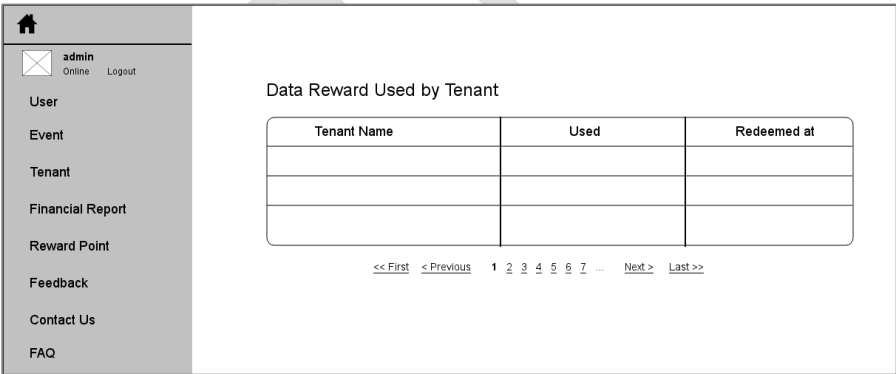
The form is titled "Edit Reward". It contains five input fields: "Reward Name" (text), "Point" (text), "Role" (dropdown menu with "tenant" selected), "Discount" (text), and "Description" (text area). At the bottom right, there are two buttons: "Close" and "Save Changes".

Gambar 3.60 Rancangan Antarmuka Mengubah Data *Reward*

Gambar 3.61 merupakan tabel yang berisi data *reward* yang dipakai *event*. Sedangkan, Gambar 3.62 merupakan tabel berisi data *reward* yang dipakai *tenant* yang terdaftar dalam CallTenant.



Gambar 3.61 Rancangan Antarmuka Data *Reward* yang Dipakai *Event*



Gambar 3.62 Rancangan Antarmuka Data *Reward* yang Dipakai *Tenant*

3.2.3.5 Rancangan Antarmuka Mengelola Pelaporan Data *Tenant*

Data *tenant* yang terdaftar dalam CallTenant ditampilkan dalam bentuk tabel yang digambarkan seperti pada Gambar 3.63.

Tenant Name	Tenant Type	Username	Start Date	End Date	Created At

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.63 Rancangan Antarmuka Data *Tenant*

Gambar 3.64 merupakan data *rating* yang didapat dari *review* bagi setiap *tenant*. Terdapat satu tombol yang berisi jumlah *rating* untuk mengakses data *review*-nya.

Tenant Name	Overall Rating & Details
nama tenant	skor rating
nama tenant	skor rating
nama tenant	skor rating

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.64 Rancangan Antarmuka Data *Review Tenant*

Gambar 3.65 merupakan data lengkap *review* untuk *tenant* dari setiap pengirim. Mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*.

Judul Review untuk Tenant

Product Quality ★★★★★

Services ★★★★★

Price ★★★★★

Quality ★★★★★

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet.

email pengirim

Gambar 3.65 Rancangan Antarmuka Detail Data *Review Tenant*

3.2.3.6 Rancangan Antarmuka Mengelola Pelaporan Data *Event*

Data *event* yang terdaftar dalam CallTenant dijabarkan dalam tabel seperti pada Gambar 3.66.

admin

Online Logout

User

Event

Tenant

Financial Report

Reward Point

Feedback

Contact Us

FAQ

Data Event

Event Name	Event Type	Username	Start Date	End Date	Created At

<< First


< Previous


1 2 3 4 5 6 7 ...

Next >

Last >>

Gambar 3.66 Rancangan Antarmuka Data *Event*





admin

Online Logout

User

Event

Tenant

Financial Report

Reward Point

Feedback

Contact Us

FAQ

Data Event Review

Event Name	Overall Rating & Details
nama event	<div> <div> <div></div> <div>skor rating</div> </div> </div>
nama event	<div> <div> <div></div> <div>skor rating</div> </div> </div>
nama event	<div> <div> <div></div> <div>skor rating</div> </div> </div>

<< First

< Previous

1

2

3

4

5

6

7

...

Next >

Last >>

Gambar 3.68 merupakan data lengkap *review* untuk *event* tersebut dari setiap pengirim, mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*.

Judul Review untuk Event	
Punctuality	★★★★★ Venue ★★★★★
Facilities & Cleanliness	★★★★★ Entertainment ★★★★★
<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet.</p>	
email pengirim	

3.2.3.7 Rancangan Antarmuka Mengelola Pelaporan Data *User*

Rancangan data *user* yang terdaftar dalam CallTenant dapat dilihat pada Gambar 3.69.

Username	Name	Email	Role	Created At	Updated At

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.69 Rancangan Antarmuka Data *User*

3.2.3.8 Rancangan Antarmuka Melihat Hasil *Feedback*

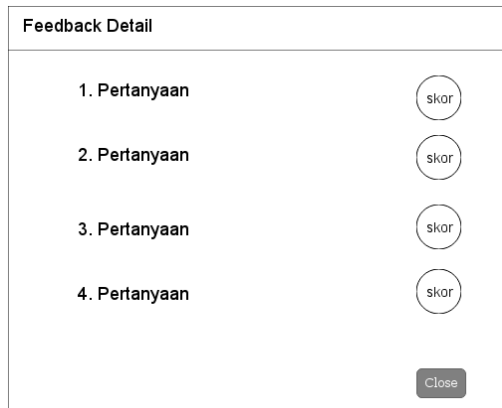
Halaman data *feedback* merupakan data *feedback* yang telah masuk ke dalam CallTenant seperti pada Gambar 3.70. Terdapat satu tombol Detail untuk melihat detail dari hasil *feedback*.

Username	Score	Date/Time	Detail
			Detail
			Detail
			Detail

<< First < Previous 1 2 3 4 5 6 7 ... Next > Last >>

Gambar 3.70 Rancangan Antarmuka Data *Feedback*

Gambar 3.71 merupakan detail hasil *feedback* beserta skor dari setiap pertanyaannya.



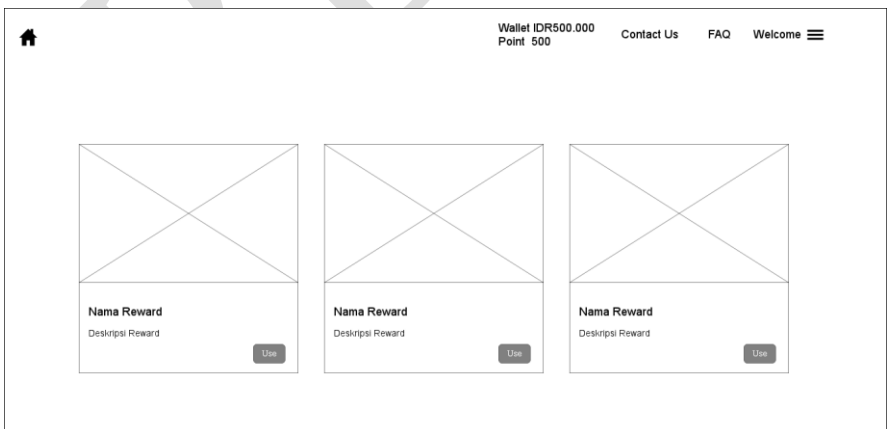
The image shows a 'Feedback Detail' form. It has a title bar 'Feedback Detail'. Below it, there are four rows, each representing a question and its score. The questions are labeled '1. Pertanyaan', '2. Pertanyaan', '3. Pertanyaan', and '4. Pertanyaan'. To the right of each question is a circular button labeled 'skor'. At the bottom right of the form is a 'Close' button.

Feedback Detail	
1. Pertanyaan	skor
2. Pertanyaan	skor
3. Pertanyaan	skor
4. Pertanyaan	skor
Close	




Gambar 3.71 Rancangan Antarmuka Detail Hasil *Feedback*

3.2.3.9 Rancangan Antarmuka Menggunakan *Reward Event*

Gambar 3.72 merupakan data *reward* yang dapat dipakai oleh *event*. Terdapat satu tombol *Use* pada setiap *reward*-nya untuk mengakses formulir pakai *reward*.



The image shows a 'Reward Event' page design. At the top, there is a header with a home icon, 'Wallet IDR500.000 Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. Below the header, there are three reward cards. Each card has a placeholder image with a large 'X' and contains the text 'Nama Reward' and 'Deskripsi Reward'. At the bottom of each card is a 'Use' button.

Wallet IDR500.000 Point 500		
Contact Us FAQ Welcome		
		

Gambar 3.72 Rancangan Antarmuka Halaman *Reward Event*

Gambar 3.73 merupakan formulir untuk menukarkan *reward* tersebut. Terdapat keterangan *point* yang dimiliki serta *balance* atau *point* yang tersisa setelah menukarkan *reward* tersebut. Pada bagian kanan bawah terdapat dua tombol yaitu *Redeem* untuk menukarkan *reward* dan *Close* untuk kembali ke data *reward*.

The image shows a web form titled "Use Reward". It contains the following information:

Jumlah Point	500
Harga Reward	300
<hr/>	
Balance	200

At the bottom right of the form are two buttons: "Redeem" and "Close".

Gambar 3.73 Rancangan Antarmuka Formulir Menggunakan *Reward Event*

3.2.3.10 Rancangan Antarmuka Menggunakan *Reward Tenant*

Gambar 3.72 merupakan data *reward* yang dapat dipakai oleh *tenant*. Terdapat satu tombol *use* pada setiap *reward*-nya untuk mengakses formulir pakai *reward*.

Sedangkan, Gambar 3.74 berisi formulir untuk menukarkan *reward* tersebut di mana diharuskan memilih *tenant* yang disediakan dalam bentuk *dropdown*. Terdapat keterangan *point* yang dimiliki serta *balance* atau *point* yang tersisa setelah menukarkan *reward* tersebut. Pada bagian kanan bawah terdapat dua tombol yaitu *Redeem* untuk menukarkan *reward* dan *Close* untuk kembali ke data *reward*.

Use Reward

Jumlah Point

500

Harga Reward

300

Balance

200

Pilih Tenant

Redeem

Close

Gambar 3.74 Rancangan Antarmuka Formulir Menggunakan *Reward Tenant*

3.2.3.11 Rancangan Antarmuka Mengisi *Feedback*

Gambar 3.75 merupakan formulir untuk mengisi *feedback*. Terdapat satu tombol *submit* untuk mengirim *feedback* tersebut seperti pada Gambar 3.75.

🏠

Wallet IDR500.000

Point 500

Contact Us

FAQ

Welcome

Feedback Form

1=Low

10=High

pertanyaan

1

2

3

4

5

6

7

8

9

10

pertanyaan

1

2

3

4

5

6

7

8

9

10

pertanyaan

1

2

3

4

5

6

7

8

9

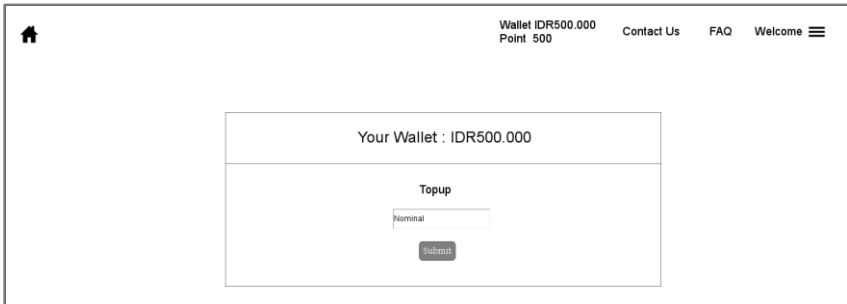
10

Submit

Gambar 3.75 Rancangan Antarmuka Formulir Mengisi *Feedback*

3.2.3.12 Rancangan Antarmuka Mengisi *Wallet*

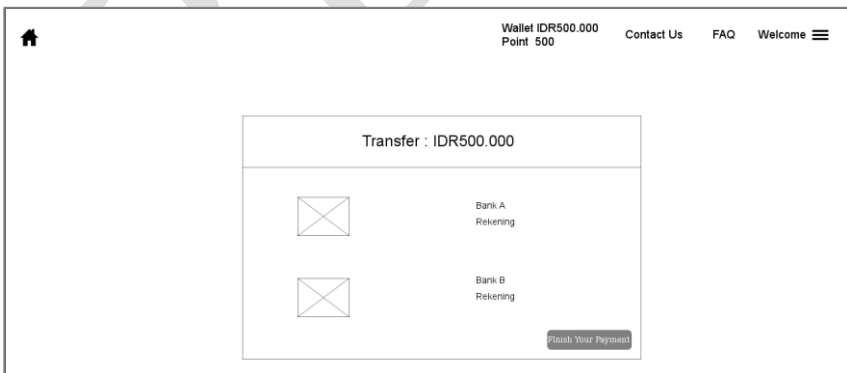
Halaman mengisi *wallet* seperti pada Gambar 3.76 merupakan formulir untuk mengisi nominal pengisian *wallet*. Terdapat satu tombol *submit* untuk mengirim formulir pengisian *wallet*.



The screenshot shows a web interface for a wallet. At the top, there is a home icon, a user balance of 'Wallet IDR500.000 Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main content area displays 'Your Wallet : IDR500.000'. Below this, there is a 'Topup' section with a 'Nominal' input field and a 'Submit' button.

Gambar 3.76 Rancangan Antarmuka Formulir Pengisian *Wallet*

Gambar 3.77 merupakan konfirmasi pembayaran yang harus diselesaikan untuk mengisi *wallet*. Terdapat nominal transfer, daftar bank yang dapat dituju, serta tombol *Finish Your Payment* untuk lanjut ke tahap berikutnya.



The screenshot shows a web interface for a wallet. At the top, there is a home icon, a user balance of 'Wallet IDR500.000 Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main content area displays 'Transfer : IDR500.000'. Below this, there are two bank options: 'Bank A Rekening' and 'Bank B Rekening', each with a checkbox. A 'Finish Your Payment' button is located at the bottom right.

Gambar 3.77 Rancangan Antarmuka Halaman Konfirmasi Pengisian *Wallet*

Gambar 3.78 merupakan formulir konfirmasi pembayaran yang harus diselesaikan untuk mengisi *wallet*. Terdapat dua tombol yaitu *Back to Payment Method* untuk kembali ke halaman sebelumnya dan *Submit Confirmation* untuk mengirim formulir tersebut.

The screenshot shows a web interface for 'Payment Confirmation'. At the top, there is a header with a home icon, 'Wallet IDR500.000', 'Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main content area contains a form with the following fields and buttons:

- ID Confirmation:** A text input field containing the value 'sdgdfwehthja'.
- Payment Date:** A date input field with a red 'x' icon and a 'Date' label.
- Receipt:** A text input field with a 'Browse...' button next to it.
- Buttons:** At the bottom of the form, there are two buttons: 'Back to Payment Method' and 'Submit Confirmation'.

Gambar 3.78 Rancangan Antarmuka Formulir Konfirmasi Pengisian *Wallet*

3.2.3.13 Rancangan Antarmuka Melihat Riwayat Konfirmasi *Topup Wallet User*

Halaman pada Gambar 3.79 merupakan data riwayat pengisian *wallet user*. Terdapat tombol Detail untuk mengakses formulir konfirmasi pembayaran yang belum diselesaikan.

The screenshot shows a web interface for 'History Confirmation'. At the top, there is a header with a home icon, 'Wallet IDR500.000', 'Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main content area contains a table with the following structure:

ID Confirmation	Date	Amount	Status	Detail
				Detail

Gambar 3.79 Rancangan Antarmuka Halaman Riwayat Konfirmasi *Topup Wallet*

3.2.3.14 Rancangan Antarmuka Melihat Riwayat Pesan *Contact Us*

Halaman pada Gambar 3.80 merupakan data riwayat pesan *Contact Us* yang pernah *user* kirimkan serta balasannya dari pihak CallTenant.

Sent Date	Response Date	Title	Questions	Answers

Gambar 3.80 Rancangan Antarmuka Halaman Riwayat Pesan *Contact Us*

3.2.3.15 Rancangan Antarmuka Melihat Riwayat Pemakaian *Point*

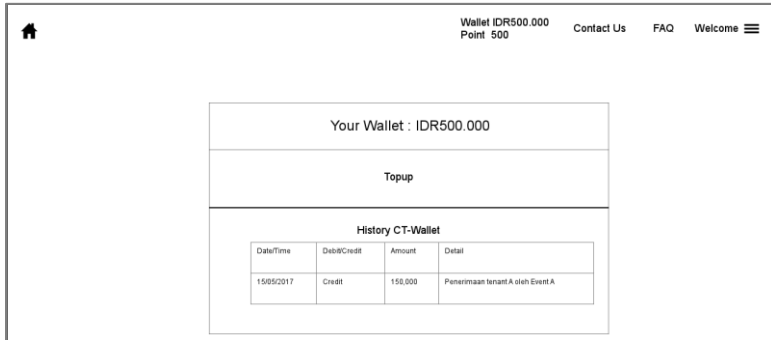
Halaman pada Gambar 3.81 merupakan data riwayat pemakaian *point user* serta keterangan pemakaiannya.

Date/Time	Debit/Credit	Amount	Detail
15/05/2017	Credit	150	Penerimaan tenant A oleh Event A

Gambar 3.81 Rancangan Antarmuka Halaman Riwayat Pemakaian *Point*

3.2.3.16 Rancangan Antarmuka Melihat Riwayat Pemakaian *Wallet*

Halaman pada Gambar 3.82 merupakan data riwayat pemakaian *wallet user* serta keterangan pemakaiannya.



Gambar 3.82 Rancangan Antarmuka Riwayat Pemakaian *Wallet*

3.2.3.17 Rancangan Antarmuka Mengelola Data *Review Tenant*

Gambar 3.83 merupakan data lengkap *review* untuk *tenant*, mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*. Terdapat dua tombol yaitu *Reply* untuk mengakses formulir balas bagi *review* tersebut dan *Show Reply* di bagian kiri untuk melihat hasil balasannya.

Judul Review untuk Tenant	
Product Quality ★★★★★	Services ★★★★★
Price ★★★★★	Quality ★★★★★
<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet.</p>	
email pengirim	
Show Reply	Reply

Gambar 3.83 Rancangan Antarmuka Halaman Data *Review Tenant*

3.2.3.18 Rancangan Antarmuka Mengelola Data *Review Event*

Gambar 3.83 merupakan data lengkap *review* untuk *event*, mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*. Terdapat dua tombol yaitu *Reply* untuk mengakses formulir balas bagi *review* tersebut dan *Show Reply* untuk melihat hasil balasannya.

Judul Review untuk Event	
Punctuality ★★★★★	Venue ★★★★★
Facilities & Cleanliness ★★★★★	Entertainment ★★★★★
<p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est. Lorem ipsum dolor sit amet.</p>	
email pengirim	
Show Reply	Reply

Gambar 3.84 Rancangan Antarmuka Halaman Data *Review Event*

3.2.3.19 Rancangan Antarmuka Menggunakan Fitur *Contact Us*

Halaman pada Gambar 3.85 merupakan formulir untuk mengirim pesan *Contact Us*. Terdapat tombol *Submit* pada bagian kanan bawah untuk mengirim pesan.

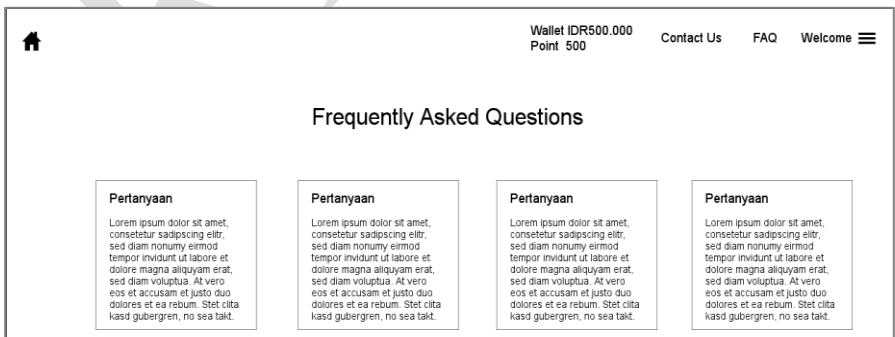


The screenshot shows a web interface for a 'Contact Us' form. At the top, there is a navigation bar with a home icon, a wallet balance of 'IDR500.000' and 'Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main heading is 'Contact Us'. Below it is a form with three input fields: 'Email', 'Message Title', and 'Message'. A 'Submit' button is located at the bottom right of the form.

Gambar 3.85 Rancangan Antarmuka Menggunakan Fitur *Contact Us*

3.2.3.20 Rancangan Antarmuka Menggunakan Fitur *Frequently Asked Questions*

Halaman pada Gambar 3.86 menampilkan data-data *Frequently Asked Questions*.

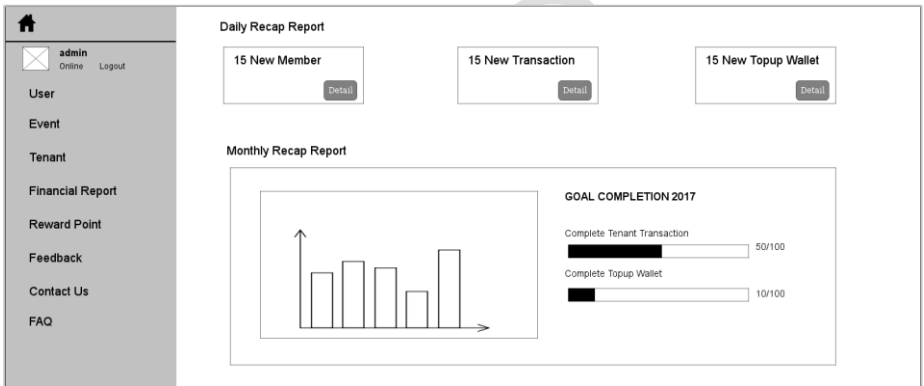


The screenshot shows a web interface for 'Frequently Asked Questions'. At the top, there is a navigation bar with a home icon, a wallet balance of 'IDR500.000' and 'Point 500', and links for 'Contact Us', 'FAQ', and 'Welcome'. The main heading is 'Frequently Asked Questions'. Below it are four cards, each titled 'Pertanyaan' and containing placeholder text: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet citta kasd gubergren, no sea takim.

Gambar 3.86 Rancangan Antarmuka Menggunakan Fitur *Frequently Asked Questions*

3.2.3.21 Rancangan Antarmuka *Dashboard Admin*

Halaman *dashboard* admin merupakan rekap singkat transaksi terbaru yang terjadi, *user* baru, hingga laporan keuangan. Terdapat grafik data keuangan dalam satu tahun serta target yang sudah dicapai yang digambarkan dengan *progress bar*. Pada bagian atas layar terdapat laporan keuangan harian yang berasal dari tiga jenis pendapatan berbeda, serta tombol Detail pada masing-masing jenis seperti pada Gambar 3.87.



Gambar 3.87 Rancangan Antarmuka *Dashboard Admin*

3.2.3.22 Rancangan Antarmuka Mengisi *Review* untuk *Tenant*

Halaman pada Gambar 3.88 merupakan formulir untuk mengisi *review* dan *rating* dari *event* untuk *tenant*. Terdapat satu tombol untuk *submit* formulir.

Write a Review for Tenant A

Choose Event to review: event 1

Product Quality: ★★★★★

Services: ★★★★★

Price: ★★★★★

Decoration: ★★★★★

Other Comments:

Submit

Gambar 3.88 Rancangan Antarmuka Mengisi *Review* untuk *Tenant*

3.2.3.23 Rancangan Antarmuka Mengisi *Review* untuk *Event*

Halaman pada Gambar 3.89 merupakan formulir untuk mengisi *review* dan *rating* dari *tenant* untuk *event*. Terdapat satu tombol untuk *submit* formulir.

Write a Review for Event A

Choose Tenant to Review: tenant 1

Punctuality: ★★★★★

Venue: ★★★★★

Facilities & Cleanliness: ★★★★★

Entertainment: ★★★★★

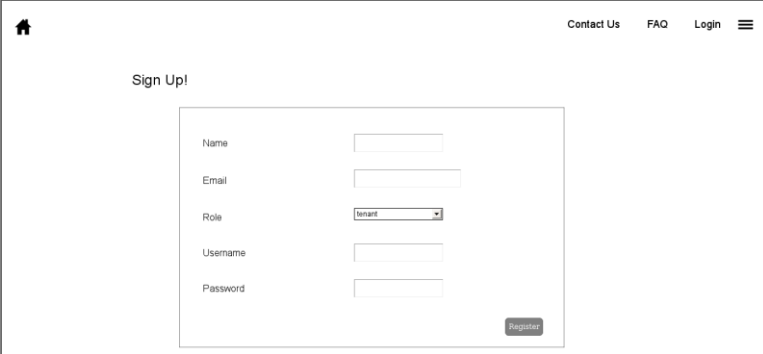
Other Comments:

Submit

Gambar 3.89 Rancangan Antarmuka Mengisi *Review* untuk *Tenant*

3.2.3.24 Rancangan Antarmuka Mendaftar sebagai *Event* atau *Tenant*

Pada halaman pendaftaran, sistem menampilkan data-data yang harus diisi oleh pengguna umum dan terdapat tombol *Register* untuk mendaftarkan diri sebagai *User* seperti pada Gambar 3.90.



The image shows a web interface for a registration page. At the top left is a home icon. At the top right are links for 'Contact Us', 'FAQ', 'Login', and a hamburger menu icon. The main heading is 'Sign Up!'. Below it is a form with five input fields: 'Name', 'Email', 'Role' (a dropdown menu with 'tenant' selected), 'Username', and 'Password'. A 'Register' button is located at the bottom right of the form.

Gambar 3.90 Rancangan Antarmuka Mendaftar Sebagai *Event* atau *Tenant*

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi adalah kode sumber untuk membangun program. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Dalam merancang aplikasi ini, digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

- Laptop
 - Tipe : Sony Vaio E-Series
 - Prosesor : Intel® Core™ i5-CPU (2.50 GHz)
 - Memori/RAM : 8GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Penjelasan perangkat lunak yang digunakan dalam implementasi aplikasi ini adalah sebagai berikut:

1. Microsoft Windows 10 Pro sebagai sistem operasi.
2. MySQL untuk mengimplementasikan rancangan basis data.
3. Sublime Text sebagai *editor* kode.

4.2 Implementasi Tampilan Antarmuka

Subbab ini membahas tentang implementasi tampilan antarmuka yang telah dirancang dan dibahas pada Bab III. Selanjutnya akan dirinci berdasarkan halaman yang akan tampil dan dilihat oleh partisipan.

4.2.1 Implementasi Mengelola Data Frequently Asked Questions

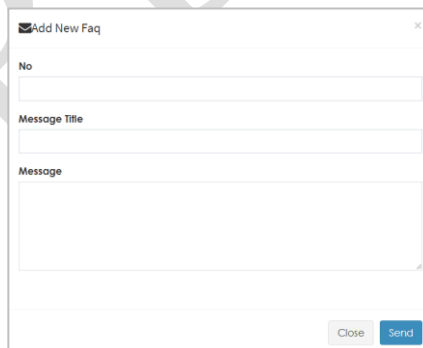
Halaman pada Gambar 4.1 menunjukkan data *Frequently Asked Questions* yang terdapat tiga tombol yaitu *Add* yang berwarna biru untuk menambah data, *update* yang berwarna hijau untuk mengubah data, serta *delete* yang berwarna merah untuk menghapus data.



No.	Title	Message	Created At	Updated At	Action
1	Cara Mendaftar di CallTenant?	Caranya: 1. Coba buka webnya 2. Klik Login, ada pilihan Join Us 3. Isi Data	07-Dec-2017 / 15.04	07-Dec-2017 / 15.04	
3	Bagaimana membuat event?	Register lalu buat	11-Dec-2017 / 04.02	07-Dec-2017 / 15.11	
6	Apa Itu Reward?	Reward adalah hadiah yang dapat kamu tukar dengan point	11-Dec-2017 / 04.01	07-Dec-2017 / 15.23	

Gambar 4.1 Implementasi Halaman *Frequently Asked Questions*

Halaman menambah data *Frequently Asked Questions* pada Gambar 4.2 merupakan formulir pengisian data serta dilengkapi dua tombol untuk *submit* yaitu *Send* yang berwarna biru dan *Close* untuk kembali ke data *Frequently Asked Questions*.



Add New Faq

No.

Message Title

Message

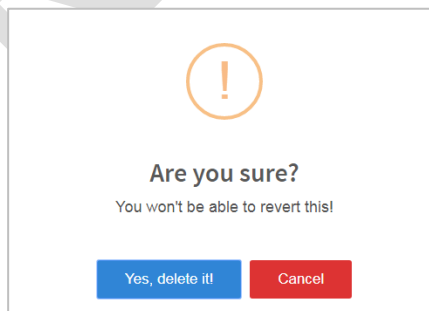
Close Send

Gambar 4.2 Implementasi Menambah Data *Frequently Asked Questions*

Halaman mengubah data *Frequently Asked Questions* pada Gambar 4.3 merupakan formulir pengisian data serta dilengkapi 2 tombol untuk *submit* yaitu *Send* yang berwarna biru dan *Close* untuk kembali ke data *Frequently Asked Questions*.

Gambar 4.3 Implementasi Mengubah Data *Frequently Asked Questions*

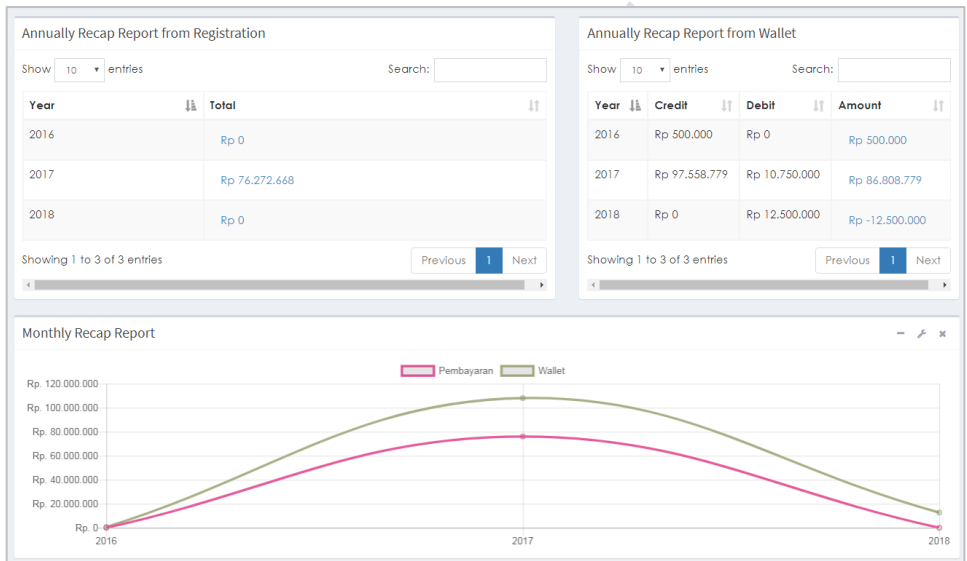
Gambar 4.4 merupakan notifikasi yang muncul saat ingin menghapus data *Frequently Asked Questions*. Terdapat dua tombol yaitu *Yes, delete it!* untuk menyetujui penghapusan data dan *Cancel* untuk menolak penghapusan data.



Gambar 4.4 Implementasi Menghapus Data *Frequently Asked Questions*

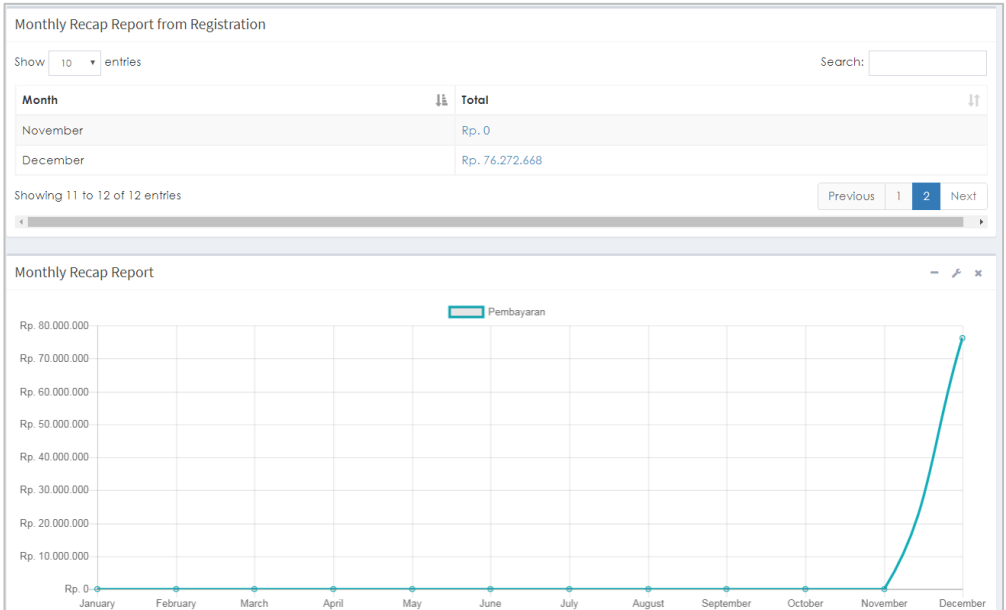
4.2.2 Implementasi Mengelola Pelaporan Keuangan

Halaman ini berisi laporan keuangan secara keseluruhan setiap tahunnya serta grafik antara pendapatan dari transaksi pendaftaran *event* dan *tenant* serta pengisian *wallet*. Pada bagian *Annually Recap Report from Registration* dan *Annually Recap Report from Wallet* terdapat tombol di setiap data total untuk mengakses laporan keuangan bulanan seperti yang dijelaskan pada Gambar 4.5.



Gambar 4.5 Implementasi Laporan Keseluruhan Laporan Keuangan

Gambar 4.6 merupakan data laporan keuangan bulanan serta grafiknya dari transaksi pendaftaran antar *event* dan *tenant*. Terdapat tombol pada setiap data total untuk mengakses data laporan keuangan harian.



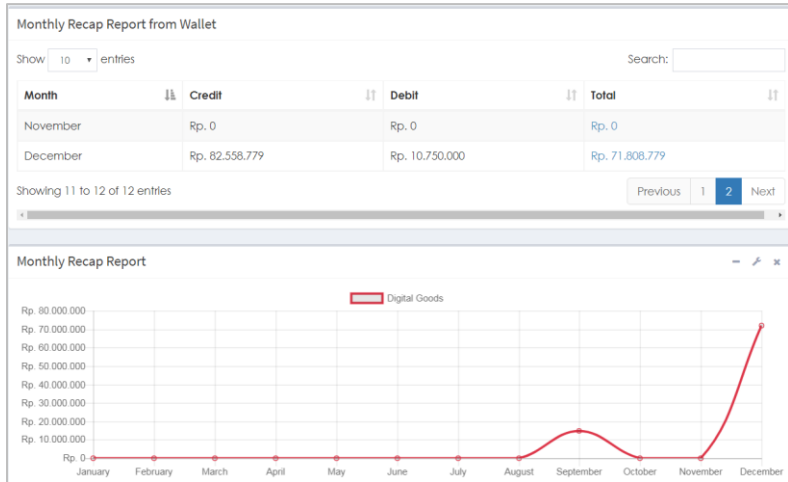
Gambar 4.6 Implementasi Laporan Bulanan Laporan Keuangan Transaksi Pendaftaran

Gambar 4.7 merupakan data laporan keuangan harian dari transaksi pendaftaran antar *event* dan *tenant*.

Daily Recap Report from Registration			
Show	10	entries	Search: <input type="text"/>
Date/Time	Total	Payment Method	
11-Dec-2017	Rp. 15.749.388	transfer	
11-Dec-2017	Rp. 104.571	transfer	
11-Dec-2017	Rp. 3.149.331	transfer	
11-Dec-2017	Rp. 15.749.181	transfer	
11-Dec-2017	Rp. 104.511	transfer	
12-Dec-2017	Rp. 15.749.071	transfer	
13-Dec-2017	Rp. 1.049.128	wallet	
13-Dec-2017	Rp. 1.049.418	transfer	
20-Dec-2017	Rp. -.356	transfer	
20-Dec-2017	Rp. 2.624.581	wallet	
Showing 1 to 10 of 16 entries			
Previous 1 2 Next			

Gambar 4.7 Implementasi Laporan Harian Laporan Keuangan Transaksi Pendaftaran

Gambar 4.8 merupakan data laporan keuangan bulanan serta grafiknya dari pengisian *wallet user*. Terdapat tombol pada setiap totalnya untuk mengakses data laporan keuangan harian.



Gambar 4.8 Rancangan Antarmuka Laporan Bulanan Laporan Keuangan Pengisian *Wallet*

Gambar 4.9 merupakan data laporan keuangan harian dari pengisian *wallet user*.

Daily Recap Report from Wallet

Show 10 entries Search:

Date/Time	Total
Credit	Rp. 200.000
Debit	Rp. 250.000
Debit	Rp. 1.000.000
Debit	Rp. 2.500.000
Debit	Rp. 5.000.000
Debit	Rp. 1.000.000
Debit	Rp. 1.000.000

Showing 21 to 27 of 27 entries Previous 1 2 3 Next

Gambar 4.9 Implementasi Laporan Harian Laporan Keuangan Pengisian *Wallet*

Halaman data transaksi *user* pada Gambar 4.10 merupakan data-data *user* yang pernah melakukan transaksi pendaftaran antar *event* dan *tenant*. Terdapat dua tombol, yaitu *Detail* yang berwarna hijau untuk melihat detail transaksi tersebut dan *Download* untuk mengunduh data menjadi format *Excel*.

Payment	Date	Total Payment	Random Number	5 %	Detail
SleepFest	2017-12-11	15749388	612	750000	Detail
SleepFest	2017-12-11	15749547	453	750000	Detail

Gambar 4.10 Implementasi Halaman Data Transaksi *User*

Gambar 4.11 merupakan detail dari setiap transaksi. Terdapat tombol *Close* untuk kembali ke data transaksi *user*.

Detail Payment

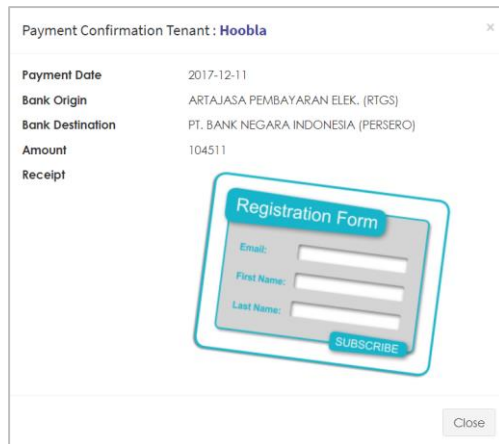
Payment for Registration Event **SleepFest** from Tenant **cuck**.

Payment Amount	15000000
Unique Code	612
5%	750000
Total Payment	15749388
Net Profit	750000 - 612 = 749388

Close

Gambar 4.11 Implementasi Halaman Detail Data Transaksi *User*

Halaman detail konfirmasi berbentuk *modal* berisi data konfirmasi pembayaran yang dilakukan setiap *tenant* seperti pada Gambar 4.12.



Payment Confirmation Tenant : **Hoobla**

Payment Date: 2017-12-11

Bank Origin: ARTAJASA PEMBAYARAN ELEK. (RTGS)

Bank Destination: PT. BANK NEGARA INDONESIA (PERSERO)

Amount: 104511

Receipt:

Registration Form

Email:

First Name:

Last Name:

SUBSCRIBE

Close

Gambar 4.12 Implementasi Detail Konfirmasi Pembayaran Transaksi User

Halaman data *wallet* pada Gambar 4.13 merupakan data-data *user* yang pernah melakukan transaksi dengan *wallet*. Terdapat dua tombol, yaitu *Detail* yang berwarna hijau untuk melihat detail penggunaan tersebut dan *Download* untuk mengunduh data menjadi format *Excel*.

Data Wallet			Download
Show	10	entries	Search: <input type="text"/>
Username	Wallet	Detail	
petrus	Rp. 12.500.000	Detail	
rifqihur11	Rp. 9.500.019	Detail	

Gambar 4.13 Implementasi Data Wallet User

Halaman konfirmasi pembayaran transaksi *user* pada Gambar 4.14 merupakan data transaksi yang perlu dikonfirmasi oleh admin setelah *tenant* melakukan pembayaran pendaftaran. Terdapat tiga tombol yaitu *accept* berwarna hijau, *delete* berwarna merah, serta Detail berwarna biru untuk melihat detail konfirmasi.

Confirmation Payment

Show 10 entries

Search:

ID Pembayaran	Payment Date	Tenant Name	Amount	Status	Action	Detail
b3041ab4-3dfc-4070-9f25-24d589acfe4	2018-01-03	Hoobla	13124364	PENDING	<div><div></div><div></div></div>	<div>Detail</div>
be7ded9f-0cf3-41db-afa0-2677cc14a5e6	2017-12-20	cuck	524524	CANCELED		<div>Detail</div>
c62db2a0-e2ae-4bcb-a7d3-81ee4da7960a	2017-12-11	Hoobla	104511	ACCEPTED		<div>Detail</div>
c7d6b964-90d6-44d9-95e2-6708e1c2edc3	2017-12-12	Pentol Gilaaa	15749134	VERIFIED		<div>Detail</div>
fcblbb67-36ec-42d7-9dc9-0d2463e7a615	0207-12-13	BakulFlanel	3149213	REJECTED		<div>Detail</div>

Showing 11 to 15 of 15 entries

Previous12Next

Gambar 4.14 Implementasi Konfirmasi Pembayaran Transaksi User

Halaman detail konfirmasi pada Gambar 4.15 berbentuk *modal* berisi data dan keterangan penggunaan *wallet* yang dilakukan setiap *tenant*.

Detail Wallet			
Date	Amount	Type	Detail
11-Dec-2017 / 08.41	10000000	Credit	Top Up
13-Dec-2017 / 07.26	20000	Credit	Top Up
<div>Close</div> <div>Send</div>			

Gambar 4.15 Implementasi Detail Data Wallet User

Halaman konfirmasi pembayaran transaksi *wallet* pada Gambar 4.16 merupakan data transaksi yang perlu dikonfirmasi oleh admin setelah *tenant* melakukan pembayaran pengisian *wallet*. Terdapat tiga tombol yaitu *accept* berwarna hijau, *delete* berwarna merah, serta Detail berwarna biru untuk melihat detail konfirmasi.

Confirmation Wallet Topup

Show

10

 entries

Search:

ID Confirmation	Username	Payment Date	Amount	Status	Action	Detail
542c9f65-a518-4047-a0b1-8dae1ea046ba	fourtrakbar	06-Dec-2017 / 00.00	90000	DONE		<div>Detail</div>
f1998c12-66e8-4021-815d-72d32d73b566	hoobla	09-Jan-2018 / 00.00	150000	IN PROGRESS	<div>✓</div> <div>✗</div>	<div>Detail</div>

Gambar 4.16 Implementasi Mengonfirmasi *Top-up Wallet*

Halaman detail konfirmasi pada Gambar 4.17 berbentuk *modal* berisi data konfirmasi pembayaran yang dilakukan setiap *tenant*.

Confirmation wallet topup user : **ISUNGGUL**
✕

Payment Date

11-Dec-2017 / 00.00

Bank Origin

PT. BANK BCA

Bank Destination

PT. BANK BCA

Amount

1000000

Receipt

INI STRUK

TRANSFER

Close

Gambar 4.17 Implementasi Detail Konfirmasi Pembayaran *Top-up Wallet*

4.2.3 Implementasi Membalas Pesan *Contact Us*

Halaman membalas pesan *Contact Us* pada Gambar 4.18 berisi data pesan yang telah masuk ke dalam aplikasi CallTenant. Terdapat dua tombol yaitu *Reply* berwarna hijau untuk mengakses formulir balas pesan dan *Detail* untuk melihat detail balasan pesan.

Incoming Messages						
Show	10	entries	Search:			
Message Title	Username	Sender Email	Date/Time	Status	Action	
Websitenya oke	sani9050	am.sani9050@gmail.com	13-Dec-2017 / 06.17	REPLIED	Detail	
tes	-	kania21@hotmail.com	18-Dec-2017 / 13.47	REPLIED	Detail	
Review	fourfrakbar	fourfr.akbar14@gmail.com	20-Dec-2017 / 03.13	NOT REPLIED	Reply	

Gambar 4.18 Rancangan Antarmuka Halaman Data *Contact Us*

Halaman formulir balas pesan pada Gambar 4.19 terdiri dari dua tombol, yaitu *Send* berwarna biru untuk *submit* formulir dan *Close* untuk kembali ke data *Contact Us*.

✉ Response Contact Us

Email

fourfr.akbar14@gmail.com

Message Title

Review

Message

Review web calltenant

Response

oke di perbaiki

Close Send

Gambar 4.19 Implementasi Membalas Pesan *Contact Us*

4.2.4 Implementasi Mengelola Data *Reward* dan *Point*

Halaman data *point* pada Gambar 4.20 merupakan data-data *user* yang pernah melakukan transaksi dengan *point*. Terdapat satu tombol Detail yang berwarna hijau untuk melihat detail penggunaan tersebut.

Data Point

Show10▼entries

Search:

Username	Point	Detail
hoobla	22523	<div>Detail</div>
tiaraanggit	13845	<div>Detail</div>

Gambar 4.20 Implementasi Data *Point* User

Gambar 4.21 merupakan detail dan keterangan penggunaan *point* user.

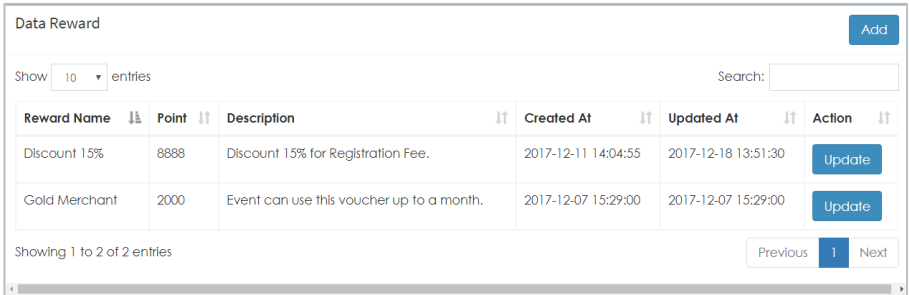
Detail Point

Date	Amount	Type	Detail
11-Dec-2017 / 08.33	2363	Credit	
11-Dec-2017 / 08.42	350	Debit	Redeem reward tenant Discount 15%
11-Dec-2017 / 08.32	20	Credit	
11-Dec-2017 / 09.07	11812	Credit	

Close

Gambar 4.21 Implementasi Detail Data *Point* User

Halaman data *reward* pada Gambar 4.22 merupakan data *reward* yang terdaftar dalam CallTenant. Terdapat dua tombol yaitu, *Add* yang berwarna biru untuk menambah data dan *Update* untuk mengubah di setiap datanya.

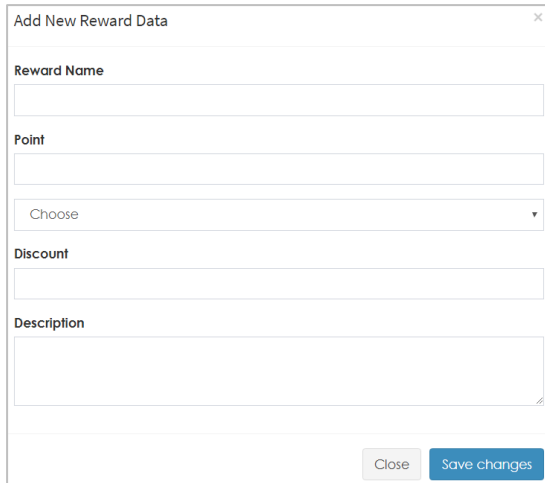


The screenshot shows a web interface titled "Data Reward". At the top right is a blue "Add" button. Below the title, there is a "Show" dropdown set to "10" and the text "entries". To the right is a "Search:" input field. The main content is a table with the following columns: "Reward Name", "Point", "Description", "Created At", "Updated At", and "Action". There are two data rows. The first row has "Discount 15%", "8888", "Discount 15% for Registration Fee.", "2017-12-11 14:04:55", and "2017-12-18 13:51:30", with an "Update" button in the "Action" column. The second row has "Gold Merchant", "2000", "Event can use this voucher up to a month.", "2017-12-07 15:29:00", and "2017-12-07 15:29:00", also with an "Update" button. Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right are "Previous", "1", and "Next" buttons.

Reward Name	Point	Description	Created At	Updated At	Action
Discount 15%	8888	Discount 15% for Registration Fee.	2017-12-11 14:04:55	2017-12-18 13:51:30	Update
Gold Merchant	2000	Event can use this voucher up to a month.	2017-12-07 15:29:00	2017-12-07 15:29:00	Update

Gambar 4.22 Implementasi Data *Reward*

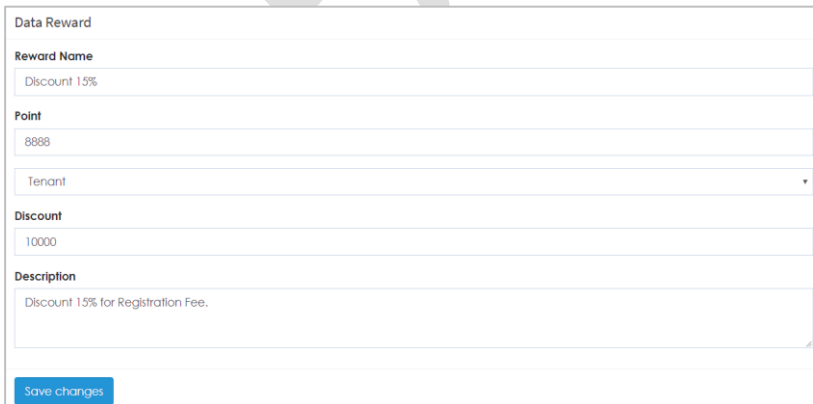
Halaman menambah data *Reward* pada Gambar 4.23 merupakan formulir pengisian data. Terdapat dua tombol untuk submit yaitu *Save Changes* yang berwarna biru dan *Close* untuk kembali ke data *Reward*.



A screenshot of a web form titled "Add New Reward Data" with a close button (X) in the top right corner. The form contains five input fields: "Reward Name" (empty), "Point" (empty), a dropdown menu labeled "Choose" (empty), "Discount" (empty), and "Description" (empty). At the bottom right, there are two buttons: "Close" (disabled) and "Save changes" (active).

Gambar 4.23 Implementasi Menambah Data *Reward*

Halaman mengubah data *Reward* pada Gambar 4.24 merupakan formulir pengisian data serta dilengkapi tombol untuk *submit* yaitu *Save Changes*.



A screenshot of a web form titled "Data Reward". The form contains five input fields: "Reward Name" (filled with "Discount 15%"), "Point" (filled with "8888"), a dropdown menu labeled "Tenant" (empty), "Discount" (filled with "10000"), and "Description" (filled with "Discount 15% for Registration Fee."). At the bottom left, there is a "Save changes" button.

Gambar 4.24 Implementasi Mengubah Data *Reward*

Gambar 4.25 merupakan data dari *reward* yang dipakai *tenant* yang terdaftar dalam CallTenant.

Data Reward Used by Tenant		
Show	10	entries
		Search: <input type="text"/>
Tenant Name	Used	Redeemed
-	not used yet	2017-12-11 08:42:45
Showing 1 to 1 of 1 entries		
		Previous 1 Next

Gambar 4.25 Implementasi Data *Reward* yang Dipakai *Tenant*

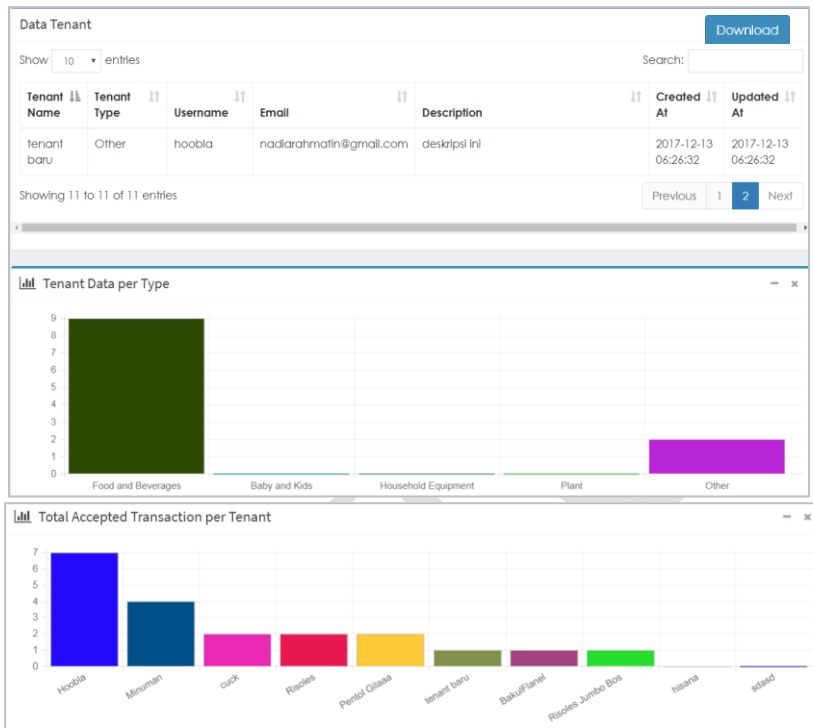
Gambar 4.26 merupakan data dari *reward* yang dipakai *event* yang terdaftar dalam CallTenant.

Data Reward Used by Event						
Show	10	entries	Search: <input type="text"/>			
Event Name	Username	Reward	Start Date	End Date	Description	Created At
Nyoman Party	rifqinur11	Gold Merchant	2017-01-01 00:00:00	2017-01-31 00:00:00	2017-12-20 06:33:05	2017-12-20 06:33:05
Reeva Schematics	schematics	Gold Merchant	2017-01-01 00:00:00	2017-01-31 00:00:00	2017-12-11 05:43:31	2017-12-11 05:43:31
SleepFest	ISUNGGUL	Gold Merchant	2017-12-01 00:00:00	2017-12-31 00:00:00	2017-12-11 09:08:51	2017-12-11 09:08:51

Gambar 4.26 Implementasi Data *Reward* yang Dipakai *Event*

4.2.5 Implementasi Mengelola Pelaporan Data *Tenant*

Halaman data *tenant* pada Gambar 4.27 merupakan daftar seluruh *tenant* serta grafik dari *tenant* yang terdaftar dalam CallTenant serta grafik *tenant* sesuai jumlah transaksi yang pernah dilakukan. Terdapat tombol *Download* berwarna biru untuk mengunduh data menjadi format *Excel*.



Gambar 4.27 Implementasi Data *Tenant*

Gambar 4.28 merupakan data *rating* yang didapat dari *review* bagi setiap *tenant*. Terdapat satu tombol yang berisi *rating* untuk mengakses data *review*-nya.

Data Tenant Review	
Show <input type="text" value="10"/> entries	Search: <input type="text"/>
Tenant Name	Overall Rating & Details
cuck	No Review
Hoobla	4.4
Pentol Gilaaa	2.3
Showing 1 to 3 of 3 entries	
Previous 1 Next	

Gambar 4.28 Implementasi Data *Review Tenant*

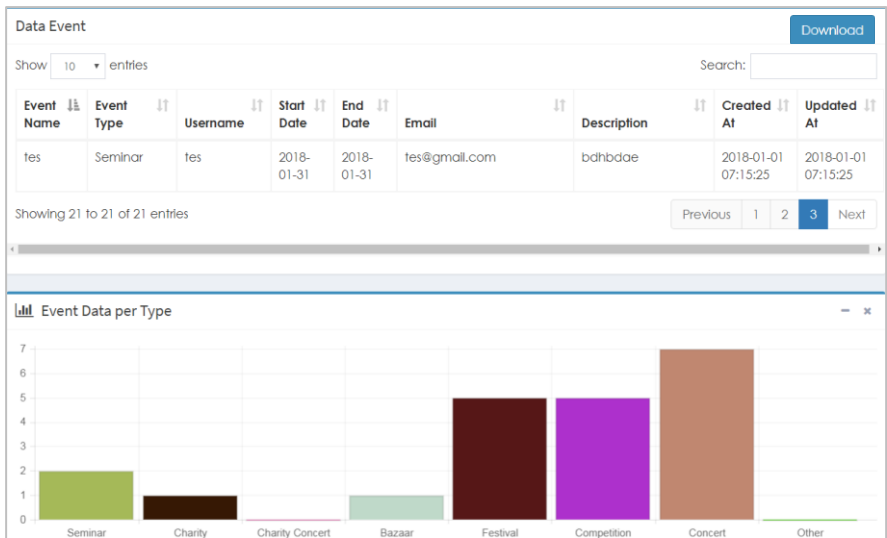
Gambar 4.29 merupakan data lengkap *review* untuk *tenant*. Mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*.

Detail Tenant Review	
Minumnya segar	
Product	★★★★★
Quality	★★★★★
Price	★★★★★
saya mencoba air putih dan itu menyenangkan	
Pentol Fest	
13-Dec-2017 / 06.12	
Show Reply	

Gambar 4.29 Implementasi Detail Data *Review Tenant*

4.2.6 Implementasi Mengelola Pelaporan Data *Event*

Halaman data *event* pada Gambar 4.30 merupakan daftar seluruh *event* serta grafik dari *event* yang terdaftar dalam CallTenant dan terdapat tombol *Download* berwarna biru untuk mengunduh data menjadi format *Excel*.



Gambar 4.30 Implementasi Data Event

Gambar 4.31 merupakan data *rating* yang didapat dari *review* bagi setiap *event*. Terdapat satu tombol yang berisi *rating* untuk mengakses data *review*-nya.

Data Event Review

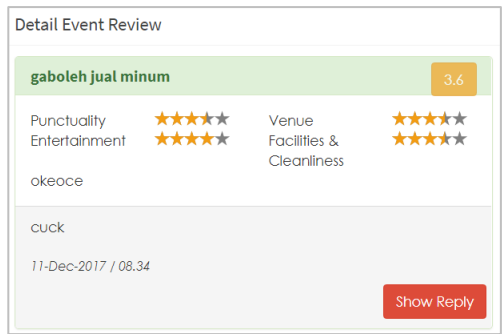
Show entries Search:

Event Name	Overall Rating & Details
Pental Fest	1.9
Reeva Schematics	3.6
SleepFest	3.9
Dance Cover	No Review
NST	No Review
INCUBUS live in Jakarta	No Review

Showing 1 to 6 of 6 entries Previous **1** Next

Gambar 4.31 Implementasi Data Review Event

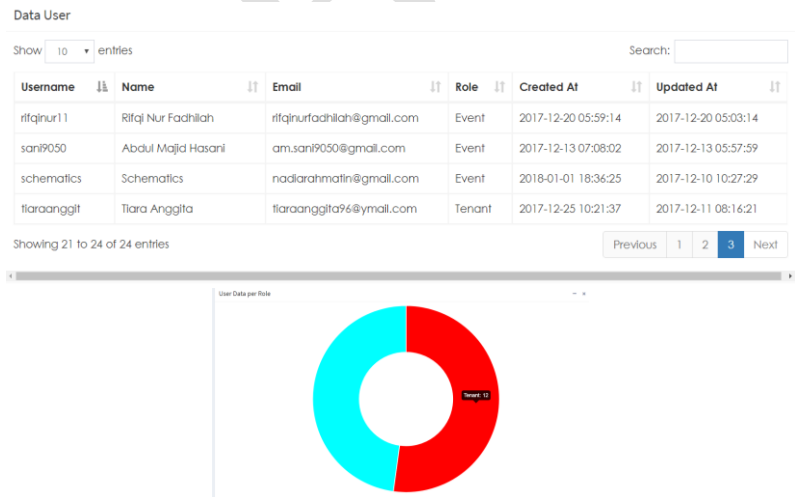
Gambar 4.32 merupakan data lengkap *review* untuk *event* tersebut, mulai dari hasil *rating* setiap elemen yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*.



Gambar 4.32 Implementasi Detail Data *Review Event*

4.2.7 Implementasi Mengelola Pelaporan Data *User*

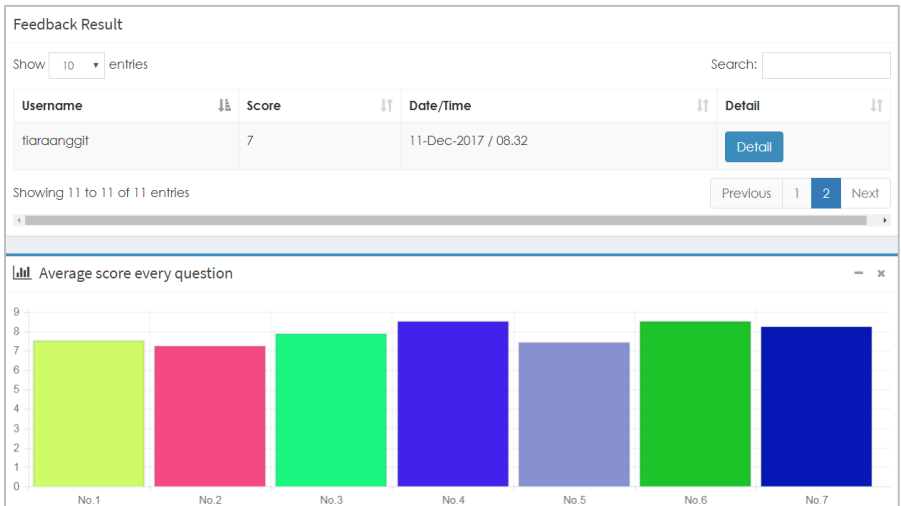
Halaman data *user* pada Gambar 4.33 merupakan daftar seluruh *user* dan grafik dari *user* yang terdaftar dalam CallTenant.



Gambar 4.33 Implementasi Data *User*

4.2.8 Implementasi Melihat Hasil *Feedback*

Halaman data *feedback* pada Gambar 4.34 berisi data dan grafik dari *feedback* yang diisi oleh *member* CallTenant. Terdapat satu tombol Detail berwarna biru untuk melihat detail dari hasil *feedback*.



Gambar 4.34 Implementasi Hasil *Feedback*

Gambar 4.35 merupakan detail hasil *feedback* yang merupakan skor setiap pertanyaan dari setiap tombol Detail user yang ditekan.

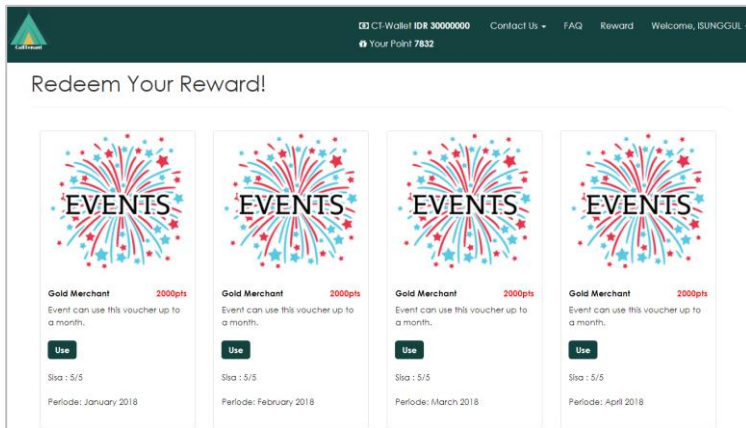
The screenshot shows a 'Feedback Detail' modal window. It contains two questions with their respective scores in blue circles:

1. How easy do you find information of event/tenant you were looking for on CallTenant? **8**
2. How much effort did you personally have to put forth to handle your request? **8**

Gambar 4.35 Implementasi Detail Hasil *Feedback*

4.2.9 Implementasi Menggunakan *Reward Event*

Halaman *reward event* pada Gambar 4.36 merupakan data *reward* yang dapat dipakai oleh *event*. Terdapat satu tombol *Use* pada setiap *reward*-nya untuk mengakses formulir pakai *reward*.



Gambar 4.36 Implementasi Halaman *Reward Event*

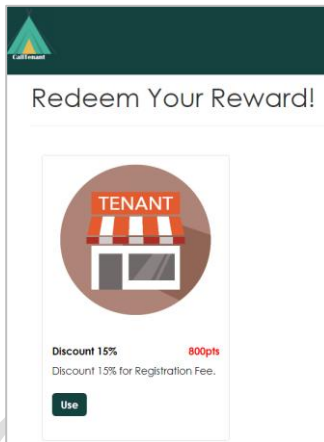
Gambar 4.37 berisi formulir untuk menukarkan *reward*. Terdapat keterangan *point* yang dimiliki serta *balance* atau *point* yang tersisa setelah menukarkan *reward* tersebut. Pada bagian kanan bawah terdapat dua tombol yaitu *Redeem* untuk menukarkan dan *Close* untuk kembali ke data *reward*.

The screenshot shows a form titled "Use Reward : Gold Merchant". It contains a table with the following data: "Your Point" with value "7832", "Reward" with value "2000" and a red minus sign "(-)" to its right, and "Event" with value "SleepFest". Below the table, there is a line for "Balance : 5832". At the bottom right, there are two buttons: "Close" and "Redeem".

Gambar 4.37 Implementasi Formulir Menggunakan *Reward Event*

4.2.10 Implementasi Menggunakan *Reward Tenant*

Halaman *reward tenant* pada Gambar 4.38 merupakan data *reward* yang dapat dipakai oleh *tenant*. Terdapat satu tombol *Use* pada setiap *reward*-nya untuk mengakses formulir pakai *reward*.



Gambar 4.38 Implementasi Halaman *Reward Tenant*

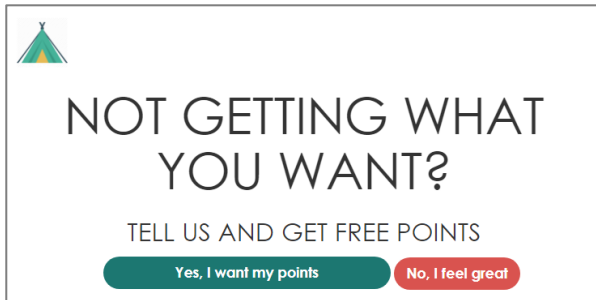
Gambar 4.39 berisi formulir untuk menukarkan *reward* tersebut. Terdapat keterangan *point* yang dimiliki serta *balance* yaitu *point* yang tersisa setelah menukarkan *reward* tersebut. Pada kanan bawah ada dua tombol yaitu *Redeem* untuk menukarkan dan *Close* untuk kembali ke data *reward*.

Use Reward : Discount 15%	
Your Point	22523
Reward	800 (-)
Choose your tenant	<input type="text" value="tenant baru ▼"/>
Balance : 21723	
Close	Redeem

Gambar 4.39 Implementasi Formulir Menggunakan *Reward Tenant*

4.2.11 Implementasi Mengisi *Feedback*

Gambar 4.40 merupakan tampilan *pop-up* untuk mengisi *feedback* akan muncul setiap kali *user* masuk halaman *profile* bagi *user* yang belum pernah mengisi *feedback*. Sedangkan, Gambar 4.41 merupakan formulir untuk mengisi *feedback*. Terdapat satu tombol *submit* untuk mengirim *feedback* tersebut.



Gambar 4.40 Implementasi *Pop-up* Mengisi *Feedback*

We'd like to check that you're completely happy with CallTenant. If you could give us 30 seconds to explain how your experience has been with us, we'd really appreciate it.

1 = Low

10 = High

1. How easy do you find information of event/tenant you were looking for on CallTenant?

1 2 3 4 5 6 7 8 9 10

2. How much effort did you personally have to put forth to handle your request?

1 2 3 4 5 6 7 8 9 10

3. How did this effort compare to your expectations?

1 2 3 4 5 6 7 8 9 10

4. Rate the overall payment methods (e.g wallet, bank transfer) out of 10?

1 2 3 4 5 6 7 8 9 10

5. Rate the overall features (e.g register, rewards and points, search tool) out of 10?

1 2 3 4 5 6 7 8 9 10

6. Do you find very useful of redeem reward feature?

1 2 3 4 5 6 7 8 9 10

7. Based on your experience, how likely would you like to recommend CallTenant to your friends, family, or colleague?

1 2 3 4 5 6 7 8 9 10

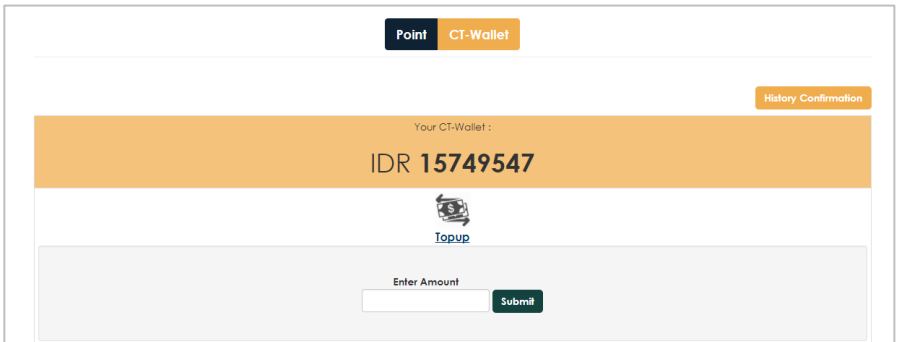
8. Please tell us the best things about CallTenant and suggest any improvements if you have them.

Submit

Gambar 4.41 Implementasi Formulir Mengisi *Feedback*

4.2.12 Implementasi Mengisi *Wallet*

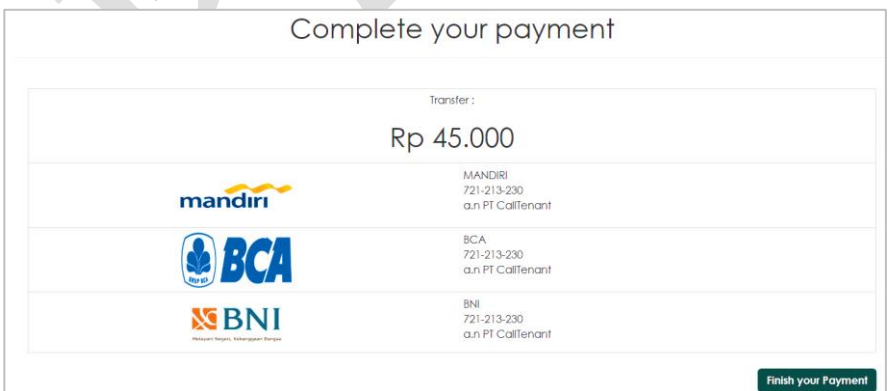
Gambar 4.42 berada pada *tab CT-Wallet* berisi formulir untuk mengisi nominal pengisian *wallet*. Terdapat dua tombol yaitu *History Confirmation* untuk melihat riwayat pengisian *wallet* dan *Submit* untuk mengirim formulir pengisian *wallet*.






The screenshot shows a web interface for the 'CT-Wallet' section. At the top, there are two tabs: 'Point' and 'CT-Wallet', with 'CT-Wallet' being the active one. On the right side, there is a button labeled 'History Confirmation'. The main content area has a header 'Your CT-Wallet :' followed by a large orange box displaying 'IDR 15749547'. Below this, there is a 'Topup' button with a dollar sign icon. Underneath, there is a form with the label 'Enter Amount' and a 'Submit' button.

Gambar 4.42 Implementasi Formulir Pengisian *Wallet*

Gambar 4.43 merupakan konfirmasi pembayaran yang harus diselesaikan untuk mengisi *wallet*. Terdapat nominal transfer, daftar bank yang dapat dituju, serta tombol *Finish Your Payment* untuk lanjut ke tahap berikutnya



The screenshot shows a payment confirmation page titled 'Complete your payment'. It displays a 'Transfer :' section with the amount 'Rp 45.000'. Below this, there is a table listing three banks: Mandiri, BCA, and BNI. Each bank entry includes its logo and the account details '721-213-230 a.n PT Calitenant'. At the bottom right, there is a button labeled 'Finish your Payment'.

Bank Logo	Account Details
	MANDIRI 721-213-230 a.n PT Calitenant
	BCA 721-213-230 a.n PT Calitenant
	BNI 721-213-230 a.n PT Calitenant

Gambar 4.43 Implementasi Halaman Konfirmasi Pengisian *Wallet*

Gambar 4.44 merupakan formulir konfirmasi pembayaran yang harus diselesaikan untuk mengisi *wallet*. Terdapat dua tombol yaitu *Back to Payment Method* untuk kembali ke halaman sebelumnya dan *Submit Confirmation* untuk mengirim formulir tersebut.

Payment Confirmation

ID Payment

b3041ab4-3dfc-4070-9

Total Payment

Rp. 13.124.364

Payment Date

dd/mm/yyyy

Bank Origin

ARTA JASA PEMBAYARAN ELEK. (RTGS)

Bank Destination

PT. BANK NEGARA INDONESIA (PERSERO)

Receipt

Choose File

No file chosen

Back to Payment Method

Submit Confirmation

Gambar 4.44 Implementasi Formulir Konfirmasi Pengisian *Wallet*

4.2.13 Implementasi Melihat Riwayat Konfirmasi *Top-up Wallet*

Halaman pada Gambar 4.45 berisi data riwayat pengisian *wallet user*. Terdapat tombol Detail untuk mengakses formulir konfirmasi pembayaran yang belum diselesaikan seperti pada Gambar 4.44.

History Confirmation

ID Confirmation	Date	Amount	Status	Detail
0ca1600d-aecc-4a91-b456-c1ddefea3695	2018-01-01 20:30:42	45000	PENDING	<div>Detail</div>

Gambar 4.45 Implementasi Halaman Riwayat Konfirmasi *Top-up Wallet*

4.2.14 Implementasi Melihat Riwayat Pesan *Contact Us*


Halaman pada Gambar 4.46 merupakan data riwayat pesan *Contact Us* yang pernah *user* kirimkan serta balasannya dari pihak CallTenant.

Response Contact Us				
Sent Date	Response Date	Title	Questions	Answers
2017-12-11 09:15:57	2017-12-11 09:16:34	Problem	Mengakses situs lambat	kami mohon maaf atas ketidak nyamanannya :{

Gambar 4.46 Implementasi Halaman Riwayat Pesan *Contact Us*

4.2.15 Implementasi Melihat Riwayat Pemakaian *Point*


Halaman pada Gambar 4.47 merupakan data riwayat pemakaian *point user* serta keterangan pemakaiannya.

 History Point			
Date/Time	Debit/Credit	Amount	Action
2017-12-11 07:38:42	Credit	79	Approval event NPC
2017-12-20 06:09:15	Credit	9844	Approval event INCUBUS live in Jakarta
2017-12-12 04:40:36	Credit	11813	Approval event SleepFest
2017-12-20 06:07:14	Credit	0	Approval event Nyoman Party
2017-12-13 06:11:50	Credit	787	Approval event Pentol Fest

Gambar 4.47 Implementasi Halaman Riwayat Pemakaian *Point*

4.2.16 Implementasi Melihat Riwayat Pemakaian *Wallet*

Halaman pada Gambar 4.48 merupakan data riwayat pemakaian *wallet user* serta keterangan pemakaiannya.




Date/Time	Debit/Credit	Amount	Detail
13-Sep-2017 / 09.07	Credit	15000000	Penerimaan <i>cuck</i> untuk SleepFest
12-Dec-2017 / 04.40	Credit	15000000	Penerimaan <i>Hoobla</i> untuk SleepFest

Gambar 4.48 Implementasi Halaman Riwayat Pemakaian *Wallet*

4.2.17 Implementasi Mengelola Data *Review Tenant*

Gambar 4.49 merupakan profil *tenant* yang berisi daftar *tenant-tenant* yang dimiliki user. Pada tabel profil terdapat tombol *rating* bergambar bintang pada kanan bawah untuk mengakses *review* bagi *tenant* tersebut.

YOUR TENANT LIST

HISTORY			
	tenant baru	ID Tenant	14352306-63eb-4dcf-82c0-28e472a35bc8
		Email	nadiahmatin@gmail.com
		Phone	0878854753
		Description	deskripsi ini
		★ 0.0	

Gambar 4.49 Implementasi Profil *Tenant*

Gambar 4.50 merupakan data lengkap *review* untuk *tenant* tersebut, mulai dari hasil *rating* setiap elemen yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*. Terdapat dua

tombol yaitu *Reply* untuk mengakses formulir balas bagi *review* tersebut dan *Show Reply* untuk melihat hasil balasannya.

Review Tenant

Minumnya segar

★ 5

Product Quality

★★★★★

Price

★★★★★

Services

★★★★★

Decoration

★★★★★

saya mencoba air putih dan itu menyenangkan

Pentol Fest

13-Dec-2017 / 06.12

Show Reply

Reply

Gambar 4.50 Implementasi Halaman Data *Review Tenant*

Gambar 4.51 merupakan formulir balas pesan yang ditujukan kepada si pengirim *review*. Terdapat tombol *Reply* untuk mengirim formulir dan *Close* untuk kembali ke data *review tenant*.

Reply review from Event : Pentol Fest

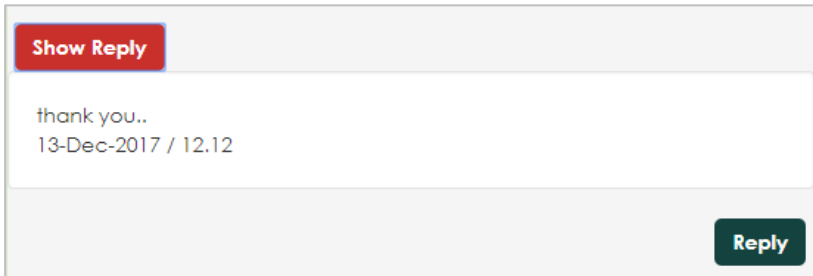
Reply Message

Close

Reply

Gambar 4.51 Implementasi Formulir Balas *Review* dari *Event* pada *Tenant*

Gambar 4.52 akan muncul saat tombol *Show Reply* ditekan dan menampilkan seluruh balasan untuk review tersebut.



Gambar 4.52 Implementasi Data Balasan *Review* dari *Event* pada *Tenant*

4.2.18 Implementasi Mengelola Data *Review Event*

Gambar 4.53 merupakan profil *event* yang berisi daftar *event-event* yang dimiliki *user*. Pada tabel profil terdapat tombol *rating* bergambar bintang pada bagian kanan untuk mengakses *review* bagi *event* tersebut.

YOUR EVENT LIST

Create New Event

[VIEW REWARD](#)

A vibrant, colorful image showing a crowd at a rave or festival with the text 'EAT SLEEP RAVE REPEAT' overlaid in bold, yellow, block letters.

SleepFest

ID Event	144dc2f5-bedb-4278-914b-2d70ea8e5496
Start Date	2017-02-02
End Date	2017-02-12
Email	team@sleepfest.com
Phone	085612349562
Location	Cibubur
Description	EAT SLEEP RAVE REPEAT!! 10 DAYS FESTIVAL!!

[Show Form](#)
[Detail Booth](#)
★ 3.9

Gambar 4.53 Implementasi Profil *Event*

Gambar 4.54 merupakan data lengkap *review* untuk *event* tersebut, mulai dari hasil *rating* setiap elemennya yang berbentuk bintang hingga deskripsi tambahan bagi si pengirim *review*. Terdapat dua tombol yaitu *Reply* untuk mengakses formulir balas bagi *review* dan *Show Reply* untuk melihat hasil balasannya.

Review Event

bagus mas

★ 3.9

Punctuality

★★★★★

Entertainment

★★★★★

Venue
Facilities &
Cleanliness

★★★★★

★★★★★

saya suka

cuck

11-Dec-2017 / 09:19

Show Reply

Reply

Gambar 4.54 Implementasi Halaman Data *Review Event*

Gambar 4.55 merupakan formulir balas pesan yang ditujukan kepada si pengirim *review*. Terdapat tombol *Reply* untuk mengirim formulir dan *Close* untuk kembali ke data *review event*.

Reply review from Tenant : cuck

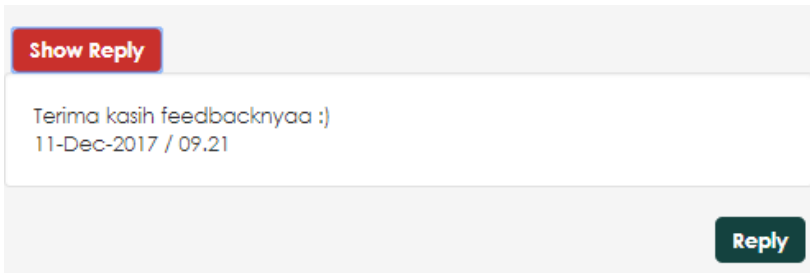
Reply Message

Close

Reply

Gambar 4.55 Implementasi Halaman Formulir Balas *Review* dari *Event* pada *Tenant*

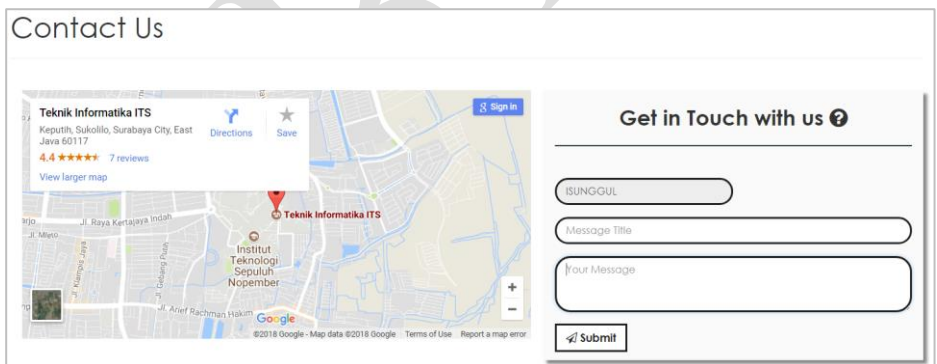
Gambar 4.56 akan muncul saat tombol *Show Reply* ditekan dan menampilkan seluruh balasan untuk review tersebut.



Gambar 4.56 Implementasi Data Balasan *Review* dari *Event* pada *Tenant*

4.2.19 Implementasi Menggunakan Fitur *Contact Us*

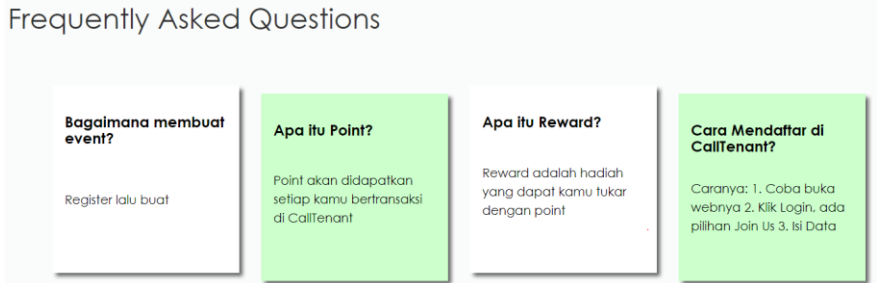
Halaman pada Gambar 4.57 merupakan formulir untuk mengirim pesan *Contact Us*. Terdapat tombol *Submit* pada bagian kiri bawah.



Gambar 4.57 Implementasi Halaman *Contact Us*

4.2.20 Implementasi Menggunakan Fitur *Frequently Asked Questions*

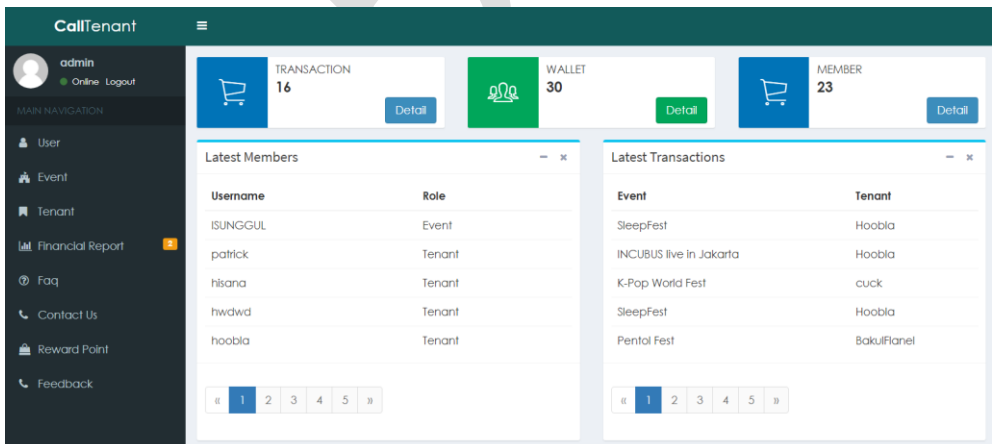
Halaman pada Gambar 4.58 menampilkan data-data *Frequently Asked Questions*.



Gambar 4.58 Implementasi Halaman *Frequently Asked Questions*

4.2.21 Implementasi Dashboard Admin

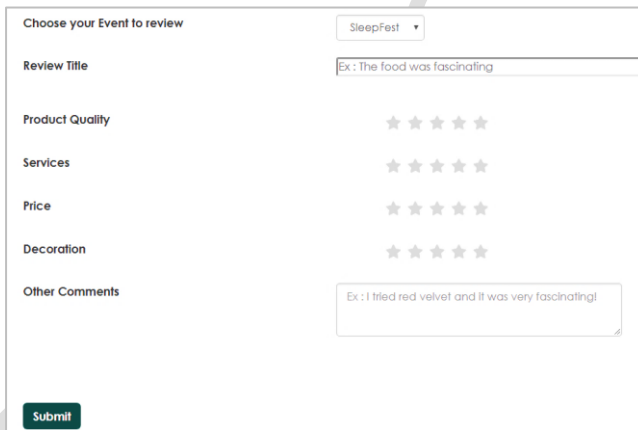
Halaman *dashboard* admin pada Gambar 4.59 berisi rekap singkat tentang jumlah transaksi yang berhasil dilakukan, *user* yang baru mendaftar, laporan keuangan, serta jumlah pemakaian *wallet*.



Gambar 4.59 Implementasi *Dashboard Admin*

4.2.22 Implementasi Mengisi *Review* untuk *Tenant*

Halaman *review* untuk *tenant* pada Gambar 4.60 merupakan formulir untuk mengisi *review* dan rating dari *event* untuk *tenant*. Dalam mengisi formulir, *user* dapat memilih *event* mana yang akan mengulas *tenant* tersebut. Saat memilih, *event* yang tersedia hanya *event* yang pernah didaftarkan pada *tenant* yang akan di *review*. Pada formulir ini terdapat fitur *rating* yang berbentuk bintang untuk setiap elemen penilaian *review tenant* yang harus diisi. Terdapat tombol untuk *Submit* formulir.



Choose your Event to review: SleepFest

Review Title: Ex: The food was fascinating

Product Quality: ★ ★ ★ ★ ★

Services: ★ ★ ★ ★ ★

Price: ★ ★ ★ ★ ★

Decoration: ★ ★ ★ ★ ★

Other Comments: Ex: I tried red velvet and it was very fascinating!

Submit

Gambar 4.60 Implementasi Mengisi *Review* untuk *Tenant*

4.2.23 Implementasi Mengisi *Review* untuk *Event*

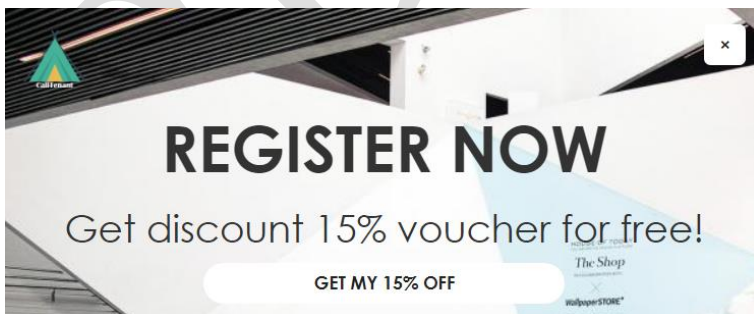
Halaman *review* untuk *event* merupakan formulir untuk mengisi *review* dan rating dari *tenant* untuk *event*. Dalam mengisi formulir, *user* dapat memilih *tenant* mana yang akan mengulas *event* tersebut. Saat memilih, *tenant* yang tersedia hanya *tenant* yang pernah mendaftar pada *event* yang akan di *review*. Pada formulir ini terdapat fitur *rating* yang berbentuk bintang untuk setiap elemen penilaian *review event* yang harus diisi. Terdapat satu tombol untuk *Submit* formulir.

The screenshot shows a web form for reviewing a tenant. At the top, there is a dropdown menu labeled 'Choose your Tenant to review' with 'Hoobla' selected. Below this is a text input field for 'Review Title' with an example: 'Ex : The food was fascinating'. The form contains five rating sections, each with five stars: 'Punctuality', 'Venue', 'Facilities & Cleanliness', and 'Entertainment'. The final section is 'Other Comments' with a text area and an example: 'Ex : I tried red velvet and it was very fascinating!'. A green 'Submit' button is located at the bottom left of the form.

Gambar 4.61 Implementasi Mengisi *Review* untuk *Event*

4.2.24 Implementasi Mendaftar sebagai *Event/Tenant*

Saat pengguna mengakses CallTenant maka dalam beberapa detik akan muncul *pop-up* seperti Gambar 4.62. Saat *user* mengklik tombol *get my 15% off* maka akan tersambung ke halaman pendaftaran, sistem menampilkan formulir seperti pada Gambar 4.63 yang harus diisi oleh pengguna umum, dan terdapat tombol *Register* untuk mendaftarkan diri sebagai *user* dengan *role event* atau *tenant*.



Gambar 4.62 Implementasi *Pop-up* Pendaftaran

Full Name

E-mail Address

Role

Username

Password (6-12 characters)

Confirm Password

REGISTER

Time Saving

No need to hesitate looking for events and tenants anymore, CallTenant provides services to complete the registration on behalf of event and tenant

Reward Point

Privilege for CallTenant member according to their respective rights

Customized Profile

Update your latest profile anytime and anywhere to attract customers

No Promotion Cost

Publish your event and tenant just for free

Gambar 4.63 Implementasi Mendaftar sebagai *Event/Tenant*

4.3 Implementasi Perangkat Lunak

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari pengambilan data *event* dan *tenant* serta pengolahan datanya.

4.3.1 Implementasi Proses Admin Mengelola Data *Frequently Asked Questions*

Implementasi proses mengelola halaman *Frequently Asked Questions* memiliki beberapa proses yang dapat dilakukan admin dari mulai melihat serta menambah, mengubah, dan menghapus data.

4.3.1.1 Proses Menambah Data *Frequently Asked Questions*

Admin menambahkan data *Frequently Asked Questions* dengan memasukkan data-data yang diperlukan menggunakan fungsi `addFaq()` yang dapat dilihat pada Kode Sumber 4.1.

1	<code>public function addFaq(Request \$request)</code>
2	<code>{</code>
3	<code> \$faq = new Faq;</code>
4	<code> \$faq->sequence = \$request->sequence;</code>

5	\$faq->id_faq = Uuid::generate(4);
6	\$faq->judul = \$request->judul;
7	\$faq->isi = \$request->isi;
8	\$faq->save();
9	}

Kode Sumber 4.1 Fungsi Menambah Data *Frequently Asked Questions*

4.3.1.2 Proses Mengubah Data *Frequently Asked Questions*

Admin mengubah data *Frequently Asked Questions* dengan **id_faq** yang sudah ada menggunakan fungsi **updateFaq()** yang dapat dilihat pada Kode Sumber 4.2.

1	public function updateFaq(Request \$request)
2	{
3	\$faq = Faq::find(\$request->id_faq);
4	\$faq->sequence = \$request->sequence;
5	\$faq->judul = \$request->judul;
6	\$faq->isi = \$request->isi;
7	\$faq->save();
8	}

Kode Sumber 4.2 Fungsi Mengubah Data *Frequently Asked Questions*

4.3.1.3 Proses Menghapus Data *Frequently Asked Questions*

Admin menghapus data *Frequently Asked Questions* dengan **id_faq** yang sudah ada menggunakan fungsi **deleteFaq()** yang dapat dilihat pada Kode Sumber 4.3.

1	public function deleteFaq(Request \$request)
2	{
3	\$faq = Faq::find(\$request->id_faq);
4	\$faq->delete();
5	}

Kode Sumber 4.3 Fungsi Menghapus Data *Frequently Asked Questions*

4.3.2 Implementasi Proses Admin Mengelola Pelaporan Keuangan

Admin dapat melihat laporan keuangan yang berasal dari transaksi pendaftaran antar *tenant* dan *event* serta pengisian *wallet* oleh *tenant* baik secara tahunan, bulanan, maupun harian. Selain itu,

admin dapat mengonfirmasi pembayaran pengisian *wallet* serta pendaftaran *tenant* pada *event*.

4.3.2.1 Proses Melihat Laporan Keuangan Keseluruhan

Pada Kode Sumber 4.4 terdapat laporan tahunan yang berasal dari pembayaran pendaftaran *tenant* yang statusnya *ACCEPTED* dan dibedakan berdasarkan tipe pembayarannya serta pembayaran pengisian *wallet* oleh *tenant*.

```
1 public function index()
2 {
3     $pembayaran = Pembayaran::with(['request'])-
4 >whereHas('request', function($query){
5     $query->where('status', 'ACCEPTED');
6     }->select(DB::raw('*,'pembayaran" as tipe'))->get();
7     $wallet = Wallet::with(['user'])->select(DB::raw('*,'wallet" as
tipe'))->get();
8
9     $merged = $pembayaran->merge($wallet);
10    $merged = $merged->sortBy('created_at');
11    $merged = $merged->mapToGroups(function($item,$key){
12        return[$item->created_at->format('Y')=>$item];
13    });
14
15    foreach ($merged as $key => $value) {
16        $merged[$key] = $merged[$key]->mapToGroups(function ($item,
$key1){
17            return [$item->tipe => $item];
18        });
19        if(!isset($merged[$key]['pembayaran']))
20        $merged[$key]['pembayaran'] = collect([]);
21
22        if(!isset($merged[$key]['wallet'])) $merged[$key]['wallet'] =
23        collect([]);
24    }
25    $merged = $merged->sortBy(function($item, $key){
26        return $key;
27    });
28    $this->data['regs'] = $merged;
```

Kode Sumber 4.4 Fungsi Melihat Laporan Keuangan Keseluruhan

4.3.2.2 Proses Melihat Laporan Keuangan Bulanan

Dari laporan tahunan sebelumnya, proses selanjutnya yang dapat dituju adalah laporan bulanan. Proses melihat laporan keuangan bulanan yang berasal dari transaksi pendaftaran *tenant* pada *event* terdapat pada Kode Sumber 4.5. Sedangkan, proses melihat laporan keuangan bulanan yang berasal dari pengisian *wallet* terdapat pada Kode Sumber 4.6.

```
1 public function regmonth(Request $request)
2 {
3     $regmonths = Pembayaran::with(['request'])-
4 >whereHas('request', function($query){
5     $query->where('status', 'ACCEPTED');
6     })->whereYear('created_at', $request->year)->get();
7
8     $regmonths = $regmonths->mapToGroups(function($item,
9 $key){
10
11         return [$item->created_at->format('F') => $item];
12     });
13     $year = $request->year;
14
15     for($i = 1; $i <= 12; $i++){
16         if(!isset($regmonths[date('F',strtotime($year.'-'.$i))]))
17 $regmonths[date('F',strtotime($year.'-'.$i))] = collect([]);
18     }
19     $regmonths = $regmonths->sortBy(function($item, $key) use($year){
20     return date('n',strtotime($key.'-'.$year));
21     });
22     $this->data['regmonths'] = $regmonths;
23     $this->data['year'] = $request->year; }
```

Kode Sumber 4.5 Fungsi Melihat Laporan Keuangan Bulanan dari Transaksi Pendaftaran

```
1 public function walmonth(Request $request)
2 {
3     $wallet = Wallet::with(['user'])->whereYear('created_at',
4 $request->year)->get();
5
6     $wallet = $wallet->mapToGroups(function($item,$key){
7         return [$item->created_at->format('F')=>$item];
8     });
9 }
```

```

7         });
8         $year = $request->year;
9         for($i = 1; $i <= 12; $i++){
10             if(!isset($wallet[date('F',strtotime($year.'-'.$i))]))
11                 $wallet[date('F',strtotime($year.'-'.$i))] = collect([]);
12             }
13             $wallet = $wallet->sortBy(function($item, $key) use($year){
14                 return date('n',strtotime($key.'-'.$year));
15             });
16             $this->data['wallets'] = $wallet;
17             $this->data['year'] = $request->year;
18         }

```

Kode Sumber 4.6 Fungsi Melihat Laporan Keuangan Bulanan dari Pengisian *Wallet*

4.3.2.3 Proses Melihat Laporan Keuangan Harian

Dari laporan bulanan sebelumnya, proses selanjutnya yang dapat dituju adalah laporan harian. Proses melihat laporan keuangan harian yang berasal dari transaksi pendaftaran *tenant* pada *event* terdapat pada Kode Sumber 4.7. Sedangkan, proses melihat laporan keuangan harian yang berasal dari pengisian *wallet* terdapat pada Kode Sumber 4.8.

```

1 public function regday(Request $request)
2 {
3     $regdays = Pembayaran::with(['request'])->whereHas('request',
4     function($query){
5         $query->where('status', 'ACCEPTED');
6         })->whereYear('created_at', $request->year)-
7     >whereMonth('created_at', date('m',strtotime($request->month.'-'.$request-
8     >year)))->get();
9     $this->data['regdays'] = $regdays;
10 }

```

Kode Sumber 4.7 Fungsi Melihat Laporan Keuangan Harian dari Transaksi Pendaftaran

```

1 public function walday(Request $request)
2 {
3     $wallet = Wallet::with(['user'])-
4     >whereMonth('created_at', date('m',strtotime($request->month.'-'.$request-
5     >year)))->whereYear('created_at', $request->year)->get();
6     $this->data['wallets'] = $wallet;

```

Kode Sumber 4.8 Fungsi Melihat Laporan Keuangan Harian dari Pengisian *Wallet*

4.3.2.4 Proses Mengonfirmasi Pembayaran Pengisian *Wallet*

Admin memiliki wewenang untuk mengonfirmasi pembayaran yang masuk ke dalam CallTenant. Pengisian *wallet* yang dilakukan *tenant* membutuhkan konfirmasi yang ditampilkan pada fungsi **confwallet()**, untuk menyetujui konfirmasi menggunakan fungsi **acceptconfwallet()**, sedangkan untuk menolak konfirmasi menggunakan fungsi **rejectconfwallet()** seperti pada Kode Sumber 4.9.

```

1  public function confwallet()
2  {
3      $this->data['confirmations'] = Confirmation::with(['user'])->get();
4  }
5  public function acceptconfwallet(Request $request)
6  {
7      $wallets = new Wallet;
8      $wallets->id_wallet = Uuid::generate(4);
9      $confirmation = Confirmation::findOrFail($request->id_confirmation);
10     $wallets->jumlah = $confirmation->jumlah;
11     $wallets->type = 'Credit';
12     $wallets->keterangan = 'Top Up';
13     $wallets->id = $confirmation->id;
14     $confirmation->id_wallet = $wallets->id_wallet;
15     $wallets->save();
16     $confirmation->status = 'DONE';
17     $confirmation->save();
18 }
19 public function rejectconfwallet(Request $request)
20 {
21     $confirmation = Confirmation::findOrFail($request->id_confirmation);
22     $confirmation->status = 'REJECTED';
23     $confirmation->save();
24 }
```

Kode Sumber 4.9 Fungsi Mengonfirmasi Pengisian *Wallet*

4.3.2.5 Proses Mengonfirmasi Pembayaran Transaksi Pendaftaran *Tenant* pada *Event*

Pembayaran konfirmasi pendaftaran pada *event* yang dilakukan *tenant* dengan fungsi **getconfPayment()**, untuk menyetujui konfirmasi menggunakan fungsi **AcceptConfPayment()**, sedangkan untuk menolak konfirmasi menggunakan fungsi **RejectConfPayment()** seperti yang dijelaskan pada Kode Sumber 4.10.

```
1 public function getconfPayment(){
2     $confirmation = DB::table('payment_confirmation')
3         ->join('pembayaran','pembayaran.id_pembayaran','=','
payment_confirmation.id_pembayaran')
4         ->join('tenant','tenant.id_tenant','=','
payment_confirmation.id_tenant')
5         ->join('request','request.id_request','=','
pembayaran.id_request')
6         -
7     >orderBy('payment_confirmation.tanggal_pembayaran','desc')->get();
8 }
9
10 public function AcceptConfPayment(Request $request){
11     DB::table('request')->where('id_request','=', $request->id_request)-
12     >update(array('status' => 'VERIFIED'));
13 }
14
15 public function RejectConfPayment(Request $request){
16     DB::table('request')->where('id_request','=', $request->id_request)-
17     >update(array('status' => 'CANCELED'));
18 }
```

Kode Sumber 4.10 Fungsi Mengonfirmasi Pendaftaran *Tenant* pada *Event*

4.3.3 Implementasi Proses Admin Membalas Pesan *Contact Us*

Pada Kode Sumber 4.11, data pesan *contact us* ditampilkan menggunakan fungsi **responseContactus()**, kemudian admin dapat membalas pesan tersebut dengan menggunakan fungsi **updateResponseContactus()**.

```

1  public function responseContactus()
2  {
3      $this->data['responses'] = Contactus::where('id',Auth::id())->get();
4  }
5
6  public function updateResponseContactus(Request $request)
7  {
8      $contactus = Contactus::find($request->id);
9      $contactus->jawaban = $request->jawaban;
10     $contactus->status = 'REPLIED';
11     $contactus->save();
12 }

```

Kode Sumber 4.11 Fungsi Membalas Pesan *Contact Us*

4.3.4 Implementasi Proses Admin Mengelola Data *Reward* dan *Point*

Implementasi proses mengelola halaman *reward* dan *point* memiliki beberapa proses yang dapat dilakukan admin dari mulai melihat serta menambah, mengubah, dan menghapus data.

4.3.4.1 Proses Menambah Data *Reward*

Admin menambahkan data *reward* dengan memasukkan data-data yang diperlukan menggunakan fungsi **create()** yang dapat dilihat pada Kode Sumber 4.12.

```

1  public function create(Request $request)
2  {
3      $reward = new Reward;
4      $reward->id_reward = Uuid::generate(4);
5      $reward->nama_reward = $request->nama_reward;
6      $reward->deskripsi = $request->deskripsi;
7      $reward->point = $request->point;
8      $reward->role = $request->role;
9      $reward->percent = $request->percent;
10     $reward->save();
11 }

```

Kode Sumber 4.12 Fungsi Menambah Data *Reward*

4.3.4.2 Proses Mengubah Data *Reward*

Admin mengubah data *reward* dengan **name** dari *reward* yang sudah ada menggunakan fungsi **updateFaq()** yang dapat dilihat pada Kode Sumber 4.13.

```
1 public function updateForm(Reward $reward)
2 {
3     $this->data['reward'] = $reward;
4     return view('admin.rewardUpdate', $this->data);
5 }
6
7 public function update(Reward $reward, Request $request)
8 {
9     $reward->nama_reward = $request->name;
10    $reward->deskripsi = $request->description;
11    $reward->point = $request->point;
12    $reward->role = $request->role;
13    $reward->percent = $request->percent;
14    $reward->save();
15 }
```

Kode Sumber 4.13 Fungsi Mengubah Data *Reward*

4.3.4.3 Proses Menghapus Data *Reward*

Admin dapat menghapus data *reward* dengan menggunakan fungsi **destroy()** yang terdapat pada Kode Sumber 4.14.

```
1 public function destroy(Reward $reward){
2     $reward->delete();
3 }
```

Kode Sumber 4.14 Fungsi Menghapus Data *Reward*

4.3.4.4 Proses Melihat *Reward* yang Dipakai *Event*

Proses melihat data *reward* yang dipakai *event*, pada Kode Sumber 4.15 diatur dalam fungsi **adminrewardevent()**. Proses ini mengambil data dari tabel **rewarduser** untuk ditampilkan.

```
1 public function adminrewardevent()
2 {
3 }
```

```

4      $rewards = RewardUser::with(['event','reward','event.user'])-
5      >whereNotNull('id_event')->whereYear('start_date',date('Y'))->get();
6
7      $rewards = $rewards->mapToGroups(function($item, $key){
8          return [$item->start_date->format('F') => $item];
9      });
10     for($i = 1; $i <= 12; $i++){
11         if(!isset($rewards[date('F',strtotime(date('Y').'-'.'$i'))]))
12         $rewards[date('F',strtotime(date('Y').'-'.'$i'))] = collect([]);
13     }
14     $rewards = $rewards->sortBy(function($item, $key){
15         return date('n', strtotime($key.'-'.'date('Y')));
16     });
17
18     $this->data['rewards'] = $rewards;
19 }

```

Kode Sumber 4.15 Fungsi Melihat Reward yang Dipakai Event

4.3.4.5 Proses Melihat Reward yang Dipakai Tenant

Proses melihat data *reward* yang dipakai *tenant* pada Kode Sumber 4.16 diatur dalam fungsi **adminrewardtenant()**. Proses ini mengambil data dari tabel **rewardtenant** untuk ditampilkan.

```

1  public function adminrewardtenant()
2  {
3      $rewards = RewardTenant::with(['reward']->whereYear('created_at',date('Y'))-
4      >get();
5      $rewards = $rewards->mapToGroups(function($item, $key){
6          return [$item->created_at->format('F') => $item];
7      });
8      for($i = 1; $i <= 12; $i++){
9          if(!isset($rewards[date('F',strtotime(date('Y').'-'.'$i'))]))
10         $rewards[date('F',strtotime(date('Y').'-'.'$i'))] = collect([]);
11     }
12     $rewards = $rewards->sortBy(function($item, $key){
13         return date('n', strtotime($key.'-'.'date('Y')));
14     });
15     $this->data['rewards'] = $rewards; }

```

Kode Sumber 4.16 Fungsi Melihat Reward yang Dipakai Tenant

4.3.4.6 Proses Melihat Data *Point*

Proses melihat data *point* user pada Kode Sumber 4.17 diambil dari tabel **point** yang ditampilkan berdasarkan **username**. Detail penggunaan *point*, *user* juga dapat dilihat berdasarkan tipenya yaitu, *Credit* dan *Debit*.

```
1 public function datapoint()
2 {
3     $points = User::with(['point'])->whereIn('role', ['Tenant', 'Event'])->get();
4     $this->data['users'] = $points->map(function($user){
5         $user->jumlah = $user->point->sum(function($point){
6             if($point->type == 'Credit') return $point->jumlah;
7             else if($point->type == 'Debit') return -($point->jumlah);
8         });
9         return $user;
10    });
11 }
```

Kode Sumber 4.17 Fungsi Melihat Data *Point*

4.3.5 Implementasi Proses Admin Mengelola Pelaporan Data *Tenant*

Admin dapat mengelola data *tenant* dengan beberapa proses yang ada dari melihat jumlah *tenant* yang terdaftar serta *review* yang dimiliki oleh *tenant* tersebut.

4.3.5.1 Proses Melihat Data *Tenant*

Proses ini melibatkan tabel **Tenant** untuk menampilkan datanya yang diatur dalam fungsi **tenant()** dan **tenantData()** karena menggunakan *datatables* dalam Kode Sumber 4.18.

```
1 public function tenant()
2 {
3     $tenants = JenisTenant::with(['tenant'])->get();
4     $this->data['tenants'] = $tenants;
5 }
6 public function tenantData()
7 {
8     $tenants = Tenant::with(['user','jenistentant'])->get();
```

9	return DataTables::of(\$Tenants)->make(true);
10	}

Kode Sumber 4.18 Fungsi Melihat Data *Tenant*

4.3.5.2 Proses Melihat Data *Review Tenant*

Proses melihat data *review tenant* terdiri dari menampilkan data *review* untuk *tenant* yang melibatkan tabel **review** dengan menyaring kolom *Sender* yang berisi *Event* diatur dalam fungsi **selectreviewTenant()** dalam Kode Sumber 4.19. Detail lengkap isi *review* untuk setiap *tenant* diatur dalam fungsi **detailreviewTenant()** pada Kode Sumber 4.20.

1	public function selectreviewTenant()
2	{
3	\$tenant = Tenant::with('review')->get();
4	
5	\$tenants = \$tenant->filter(function (\$value, \$key) {
6	return !\$value->review->isEmpty();
7	});
8	
9	return DataTables::of(\$tenants)
10	->addColumn('total', function(\$tenant){
11	\$tenanttotalrating = 0;
12	\$jumlahreviewpertenant = 0;
13	foreach (\$tenant->review->where('sender', 'Event') as \$key =>
14	\$value) {
15	\$tenanttotalrating += ((\$value->kualitas_produk_tenant +
16	\$value->pelayanan_tenant + \$value->harga_tenant + \$value->
17	dekor_tenant) / 4);
18	\$jumlahreviewpertenant++;
19	}
20	if (\$jumlahreviewpertenant == 0) return 'No Review';
21	return '
22	id_tenant.'"><button class="btn btn-primary" style="text-align:center">'.number_format(\$tenanttotalrating/\$jumlahreviewpertenant,
23	1).'</button>';
24	})
25	->rawColumns(['total'])
26	->make(true);
27	}

Kode Sumber 4.19 Fungsi Melihat *Review* untuk *Tenant*

1	public function detailreviewTenant(Request \$request)
2	{
3	\$this->data['reviews'] = Review::with(['tenant'])
4	->where('id_tenant',\$request->id)->where('sender','Event')->get();
5	}

Kode Sumber 4.20 Fungsi Melihat Detail Review untuk Tenant

4.3.6 Implementasi Proses Admin Mengelola Pelaporan Data *Event*

Admin dapat mengelola data *event* dengan beberapa proses yang ada, dari melihat jumlah *event* yang terdaftar, serta *review* yang dimiliki oleh *event* tersebut.

4.3.6.1 Proses Melihat Data Event

Proses ini melibatkan tabel **event** untuk menampilkan datanya yang diatur dalam fungsi **event()** dan **eventData()** karena menggunakan *datatables* dalam Kode Sumber 4.21.

1	public function event()
2	{
3	\$events = JenisEvent::with(['event'])->get();
4	\$this->data['events'] = \$events;
5	return view('admin.event', \$this->data);
6	}
7	
8	public function eventData()
9	{
10	\$events = Event::with(['user', 'jenisevent'])->get();
11	return DataTables::of(\$events)->make(true);
12	}

Kode Sumber 4.21 Fungsi Melihat Data Event

4.3.6.2 Proses Melihat Data Review Event

Proses melihat data *review event* terdiri dari menampilkan data *review* untuk *event* yang melibatkan tabel **review** dengan menyaring kolom *Sender* yang berisi *Tenant* diatur dalam fungsi **selectreviewEvent()** pada Kode Sumber 4.22. Detail lengkap isi

review untuk setiap *event* diatur dalam fungsi **detailreviewEvent()** pada Kode Sumber 4.23.

```
1 public function selectreviewEvent()  
2 {  
3     $reviewevents = Review::with  
4     (['review']->get());  
5  
6     $sevents = $event->filter(function($value,$key){  
7         return !$value->review->isEmpty();  
8     });  
9  
10    return DataTables::of($sevents)  
11    ->addColumn('total', function($event){  
12        $eventtotalrating = 0;  
13        $jumlahreviewperevent = 0;  
14  
15        foreach( $event->review as $key => $value)  
16        {  
17            $eventtotalrating += (($value->ketepatan_waktu_event +  
18            $value->venue_event + $value->hiburan_event + $value->fasilitas_event)  
19            / 4);  
20            $jumlahreviewperevent++;  
21        }  
22  
23        return '<a href="'.route('detailreviewevent').'?id='.$event->  
>id_event.'"><button class="btn btn-primary" style="text-align:center">'.number_format($eventtotalrating/$jumlahreviewperevent,  
1).'/</button></a>';  
24    })  
25    ->rawColumns(['total'])  
26    ->make(true);  
27 }
```

Kode Sumber 4.22 Fungsi Melihat Review untuk Event

```
1 public function detailreviewEvent(Request $request)  
2 {  
3     $this->data['reviews'] = Review::with(['event'])  
4     ->where('id_event',$request->id)->where('sender','Tenant')->get();  
5 }
```

Kode Sumber 4.23 Fungsi Melihat Detail Review untuk Event

4.3.7 Implementasi Proses Admin Mengelola Pelaporan Data User

Admin dapat mengelola data *user* dengan beberapa proses yang ada dengan melihat jumlah *user* yang terdaftar. Proses ini melibatkan tabel **user** untuk menampilkan datanya yang diatur dalam fungsi **adminuser()** dan **userData()** karena menggunakan *datatables* serta fungsi **chart()** untuk mengatur data yang ditampilkan pada grafik dalam Kode Sumber 4.24.

```
1 public function adminuser()
2 {
3     $this->data['users'] = User::get();
4 }
5
6 public function userData()
7 {
8     $users = User::get();
9     return Datatables::of($users)
10        ->make(true);
11 }
12 public function chart()
13 {
14     $users = collect(Auth::id());
15     $users->count();
16 }
```

Kode Sumber 4.24 Fungsi Melihat Data *User*

4.3.8 Implementasi Proses Admin Melihat Hasil *Feedback*

Proses pada Kode Sumber 4.25 digunakan untuk melihat hasil *feedback* terhadap CallTenant berikut skornya yang dimasukkan oleh *event* dan *tenant* diatur dalam fungsi **admin()**.

```
1 public function admin()
2 {
3     $this->data['feedbacks'] = Feedback::with('user')->get();
4 }
```

Kode Sumber 4.25 Fungsi Melihat Hasil *Feedback*

4.3.9 Implementasi Proses *Event* Menggunakan *Reward Event*

Proses menggunakan *reward* bagi *event* pada Kode Sumber 4.26 diatur dalam fungsi **useReward()** yang melibatkan tiga tabel sekaligus yaitu, **reward**, **point**, dan **rewarduser**. Saat *event* menukarkan *reward* maka **id_reward** dicari dalam tabel **reward**. Kemudian, **id_point** baru dibuat pada tabel **point** untuk proses penambahan *point* dan **id_rewarduser** baru dibuat pada tabel **rewarduser** untuk menjadi tanda pemakaian *reward*.

```
1 public function useReward(Event $event, Request $request)
2 {
3     $reward = Reward::findOrFail($request->id_reward);
4
5     if(Auth::user()->getPoint() - $reward->point < 0)
6     return back()->with('status', 'Your Point is insufficient');
7
8     // Tabel Point
9     $point = new Point;
10    $point->id_point = Uuid::generate(4);
11    $point->id = Auth::id();
12    $point->id_event = $event->id_event;
13    $point->type = 'Debit';
14    $point->keterangan = 'Beli reward';
15    $point->jumlah = $reward->point;
16    $point->id_reward = $reward->id_reward;
17    $point->save();
18    // Tabel Reward User
19    $user = new RewardUser;
20    $user->id_rewarduser = Uuid::generate(4);
21    $user->id = Auth::id();
22    $user->id_event = $event->id_event;
23    $user->id_reward = $request->id_reward;
24    $user->start_date = Carbon::create(date('Y'), $request-
>bulanAngka, 1, 0, 0, 0);
25    $user->end_date = Carbon::create(date('Y'), $request-
>bulanAngka, date("t", strtotime($user->start_date->toDateString())),
0, 0, 0);
26    $user->save();
27 }
28
```

Kode Sumber 4.26 Fungsi *Event* Menggunakan *Reward*

4.3.10 Implementasi Proses *Tenant* Menggunakan *Reward* *Tenant*

Proses menggunakan *reward* bagi *tenant* pada Kode Sumber 4.27 diatur dalam fungsi **useRewardTenant()** yang melibatkan tiga tabel sekaligus yaitu, **reward**, **point**, dan **rewardtenant**. Saat *event* menukarkan *reward* maka **id_reward** dicari dalam tabel **reward**. Kemudian, **id_point** baru dibuat pada tabel **point** untuk proses penambahan *point* dan **id_rewardtenant** baru dibuat pada tabel **rewardtenant** untuk menjadi tanda pemakaian *reward*.

```
1 public function useRewardTenant(Request $request)
2 {
3     $reward = Reward::findOrFail($request->id_reward);
4
5     //Tabel Point
6     $point = new Point;
7     $point->id_point = Uuid::generate(4);
8     $point->id = Auth::id();
9     if(Auth::user()->role == 'Tenant') $point->id_tenant = $request-
>id_tenant;
10    $point->type = 'Debit';
11    $point->jumlah = $reward->point;
12    $point->keterangan = 'Beli reward';
13    $point->id_reward = $reward->id_reward;
14    $point->save();
15
16    //Tabel RewardTenant
17    $user = new RewardTenant;
18    $user->id_rewardtenant = Uuid::generate(4);
19    $user->id = Auth::id();
20    $user->flag = 0;
21    $user->id_reward = $reward->id_reward;
22    $user->save();
23 }
```

Kode Sumber 4.27 Fungsi *Tenant* Menggunakan *Reward*

4.3.11 Implementasi Proses *Event* Mendapatkan *Point*

Proses *event* mendapatkan *point* akan terjadi saat *event* mengonfirmasi pendaftaran dari *tenant*-nya. Jika *event* menyetujui pendaftaran maka menggunakan fungsi **AcceptTenant()** dan fungsi

ini akan menambahkan data ke tabel **point** dengan jumlah setengah dari point bagi tenant.

```
1 public function AcceptTenant(Request $request){
2     DB::table('request')->where('id_request','=', $request->id_request)
3     ->update(array('status' => 'ACCEPTED'));
4     $status = DB::table('pembayaran')
5     ->where('id_request','=', $request->id_request)->get();
6     $id_event = DB::table('event')
7         ->join('request','request.id_event','=', 'event.id_event')
8         ->where('id_request','=', $request->id_request)->get();
9     $id_tenant = DB::table('tenant')
10        ->join('request','request.id_tenant','=', 'tenant.id_tenant')
11        ->where('id_request','=', $request->id_request)->get();
12    //Point untuk Event
13    $point1 = new Point;
14    $point1->id_point = Uuid::generate(4);
15    $point1->id = $id_event[0]->id;
16    $point1->id_tenant = $id_tenant[0]->id_tenant;
17    $point1->id_event = $id_event[0]->id_event;
18    $point1->type = 'Credit';
19    $point1->keterangan = 'Approval';
20
21    //Point untuk Tenant
22    $point = new Point;
23    $point->id_point = Uuid::generate(4);
24    $point->id = $id_tenant[0]->id;
25    $point->id_tenant = $id_tenant[0]->id_tenant;
26    $point->type = 'Credit';
27    $point->keterangan = 'Approval';
28    $point->id_event = $id_event[0]->id_event;
29    $point->jumlah = $status[0]->jumlah_point;
30    $point->save();
31
32    $point1->jumlah = (int) $status[0]->jumlah_point/2;
33    $point1->save();
32 }
```

Kode Sumber 4.28 Fungsi Event Mendapatkan Point

4.3.12 Implementasi Proses Tenant Mendapatkan Point

Proses ini terjadi saat pendaftaran *tenant* disetujui oleh *event*, seperti dalam Kode Sumber 4.28. Jumlah *point* terbentuk dari dua

jenis pembayaran yang berbeda ,yaitu *point* dari pembayaran melalui transfer bank yang diatur dalam fungsi **pointTransfer()** dan *point* dari pembayaran melalui *wallet* yang diatur dalam fungsi **pointWallet()**.

```
1 public function pointTransfer()
2 {
3     $pembayaran = Pembayaran::findorFail($request->id_pembayaran);
4     $multi = PointMultiplier::where('nama','transfer')->firstOrFail();
5     $point = new Point;
6     $point->id_point = Uuid::generate(4);
7     $point->jumlah = $multi->jumlah*$point->jumlah;
8     $point->id = Auth::id();
9     $point->type = 'Credit';
10    $point->save();
11    return view('user.pembayaranconf');
12 }
13
14 public function pointWallet()
15 {
16     $pembayaran = Pembayaran::findorFail($request->id_pembayaran);
17     $multi = PointMultiplier::where('nama','wallet')->firstOrFail();
18     $point = new Point;
19     $point->id_point = Uuid::generate(4);
20     $point->jumlah = $multi->jumlah*$point->jumlah;
21     $point->id = Auth::id();
22     $point->type = 'Credit';
23     $point->save();
24     return view('user.pembayaranconf');
25 }
26 }
```

Kode Sumber 4.29 Fungsi Tenant Mendapatkan Point

4.3.13 Implementasi Proses User Mengisi *Feedback*

Proses pada Kode Sumber 4.30 digunakan saat *user* akan mengisi *feedback*. Formulir ditampilkan dengan fungsi **viewfeedback()** dan pengisian formulirnya diatur dengan fungsi **addfeedback()**.

```
1 public function viewfeedback(){
2     return view('user.feedback');
3 }
```

```

4  public function addfeedback(Request $request)
5  {
6      $feedback = new Feedback;
7      $feedback->id_feedback = Uuid::generate(4);
8      $feedback->id = Auth::id();
9      $feedback->no1 = $request->no1;
10     $feedback->no2 = $request->no2;
11     $feedback->no3 = $request->no3;
12     $feedback->no4 = $request->no4;
13     $feedback->no5 = $request->no5;
14     $feedback->no6 = $request->no6;
15     $feedback->no7 = $request->no7;
16     $feedback->no8 = $request->no8;
17     $feedback->total = ($request->no1 + $request->no2 + $request-
>no3 + $request->no4 + $request->no5 + $request->no6 + $request-
>no7)/7;
18     $feedback->save();
19     $point = new Point;
20     $point->id_point = Uuid::generate(4);
21     $point->id = Auth::id();
22     $point->jumlah = 20;
23     $point->type = 'Credit';
24     $point->keterangan = 'Feedback Reward';
25     $point->save();
26 }

```

Kode Sumber 4.30 Fungsi User Mengisi Feedback

4.3.14 Implementasi Pop-up untuk User Mengisi Feedback

Proses *user* mengisi *feedback* ditampilkan sebelumnya kepada pengguna dengan *pop-up* seperti dalam Kode Sumber 4.31. Fungsi ini akan mengarahkan user ke proses mengisi *feedback* pada Kode Sumber 4.30.

```

1  @if(Auth::user()->feedback->isEmpty())
2  <script type="text/javascript">
3  $(document).ready(function(){
4      setTimeout(function(){
5          $('#myModal').modal('toggle'); }, 3000);});
6  </script>
7  @endif

```

Kode Sumber 4.31 Implementasi Pop-up untuk User Mengisi Feedback

4.3.15 Implementasi Proses *Tenant* Mengisi *Wallet*

Proses mengisi *wallet* diatur dengan fungsi **addtopupwallet()**. Saat *tenant* selesai mengisi *wallet* maka akan terbentuk data baru pada tabel **confirmation** yang terdapat pada Kode Sumber 4.32. Setelah data konfirmasi terbentuk maka *tenant* harus mengisi data konfirmasi yang diatur dengan fungsi **updatetopupwallet()** dalam Kode Sumber 4.33.

```
1 public function addtopupwallet(Request $request)
2 {
3     if($request->has('wallets')){
4         $this->data['wallets'] = Confirmation::findOrFail($request-
>wallets);
5         if($this->data['wallets']->id != Auth::id()) abort(404);
6     }
7     else{
8         $wallets = new Confirmation;
9         $wallets->id_confirmation = Uuid::generate(4);
10        $wallets->id = Auth::id();
11        $wallets->jumlah = $request->jumlah;
12        $wallets->status = 'PENDING';
13        $wallets->save();
14        $this->data['wallets'] = $wallets;
15    }
```

Kode Sumber 4.32 Fungsi Menambah Data *Wallet*

```
1 public function updatetopupwallet(Request $request)
2 {
3     $wallets = Confirmation::findOrFail($request->id_confirmation);
4     $wallets->tgl = $request->tgl;
5     $wallets->rek = $request->rek;
6     $wallets->bank_asal = $request->bank_asal;
7     $wallets->bank_tujuan = $request->bank_tujuan;
8     $wallets->status = 'IN PROGRESS';
9     if($request->hasFile('receipt')){
10
11         $photo = $request->receipt;
12         $a = $photo->getClientOriginalName();
13         $filename = time().'.'.$a;
14         $photos = $photo-
>move(public_path('/uploads/bukti/'),$filename);
```

15	<code>\$wallets->path = 'uploads/bukti/'.\$filename;</code>
16	<code>}</code>
17	<code>\$wallets->save();</code>
18	<code>return back()->with('status', "Your confirmation is being</code>
19	<code>processed");</code>
20	<code>}</code>

**Kode Sumber 4.33 Fungsi Mengisi Konfirmasi Pembayaran Pengisian
Wallet**

4.3.16 Implementasi Proses *Tenant* Melihat Riwayat Konfirmasi *Topup Wallet*

Proses melihat riwayat konfirmasi *topup wallet* pada Kode Sumber 4.34 diatur dengan fungsi **history()** yang mengambil data dari tabel **confirmation**.

1	<code>public function history()</code>
2	<code>{</code>
3	<code> \$this->data['wallets'] = Confirmation::with(['user'])-</code>
4	<code>>where('id',Auth::id())->get();</code>
5	<code>}</code>

Kode Sumber 4.34 Fungsi *Tenant* Melihat Riwayat Konfirmasi *Topup Wallet*

4.3.17 Implementasi Proses User Melihat Riwayat Pesan *Contact Us*

Proses melihat riwayat pesan *Contact Us* pada Kode Sumber 4.35 diatur dengan fungsi **responseContactus()** yang mengambil data dari tabel **contactus**.

1	<code>public function responseContactus()</code>
2	<code>{</code>
3	<code> \$this->data['responses'] = Contactus::where('id',Auth::id())->get();</code>
4	<code>}</code>
5	

Kode Sumber 4.35 Fungsi User Melihat Riwayat Pesan *Contact Us*

4.3.18 Implementasi Proses *User* Melihat Riwayat Pemakaian *Point*

Proses melihat riwayat *point* pada Kode Sumber 4.36 diatur dengan fungsi **point()** untuk mengakses datanya dan fungsi **getPoint()** mengatur detail penggunaan *point* tersebut.

```
1 public function point()
2 {
3     return $this->hasMany('App\Point', 'id');
4 }
5
6 public function getPoint()
7 {
8     return Auth::user()->point->sum(function($item){
9         if($item->type == 'Credit'){
10             return $item->jumlah;
11         }
12         elseif ($item->type == 'Debit') {
13             return -$item->jumlah;
14         }
15     });
16 }
```

Kode Sumber 4.36 Fungsi *User* Melihat Riwayat Pemakaian *Point*

4.3.19 Implementasi Proses *User* Melihat Riwayat Pemakaian *Wallet*

Proses melihat riwayat *wallet* pada Kode Sumber 4.37 diatur dengan fungsi **wallet()** untuk mengakses datanya dan fungsi **getWallet()** mengatur detail penggunaan *wallet* tersebut.

```
1 public function wallet()
2 { return $this->hasMany('App\Wallet', 'id'); }
3 public function getWallet()
4 {
5     return Auth::user()->wallet->sum(function($item){
6         if($item->type == 'Credit'){
7             return $item->jumlah;
8         }
9         elseif ($item->type == 'Debit') {
10             return -$item->jumlah;
11         }
12     });
13 }
```

11	}
12	});
13	}

Kode Sumber 4.37 Fungsi *User* Melihat Riwayat Pemakaian *Wallet*

4.3.20 Implementasi Proses *Tenant* Mengelola Data *Review Tenant*

Proses mengelola data *review tenant* pada Kode Sumber 4.38 terdiri dari menampilkan data *review*-nya yang diatur dengan fungsi **reviewnyaTenant()** dan *tenant* juga dapat membalas setiap *review*-nya dengan fungsi **replyreviewnyaTenant()**.

1	public function reviewnyaTenant(Request \$request)
2	{
3	\$this->data['reviews'] = Review::with(['tenant','reply'])
4	->where('id_tenant',\$request->id_tenant)->where('sender','Event')->get();
5	}
6	
7	public function replyreviewnyaTenant(Request \$request)
8	{
9	\$review = Review::findOrFail(\$request->id_review);
10	\$id_tenant = (\$request->id_tenant);
11	\$reply = new ReplyReview;
12	\$reply->id = Uuid::generate(4);
13	\$reply->id_review = \$request->id_review;
14	\$reply->isi = \$request->isi;
15	\$reply->save();
16	}

Kode Sumber 4.38 Fungsi *Tenant* Mengelola Data *Review Tenant*

4.3.21 Implementasi Proses *Event* Mengelola Data *Review Event*

Proses mengelola data *review event* pada Kode Sumber 4.39 terdiri dari menampilkan data *review*-nya yang diatur dengan fungsi **reviewnyaEvent()** dan *event* juga dapat membalas setiap *review*-nya dengan fungsi **replyreviewnyaEvent()**.

1	public function reviewnyaEvent(Request \$request)
2	{
3	\$this->data['reviews'] = Review::with(['event'])

```

4  ->where('id_event',$request->id_event)->where('sender','Tenant')->get();
5  }
6
7  public function replyreviewnyaEvent(Request $request)
8  {
9      $review = Review::findOrFail($request->id_review);
10     $id_event = ($request->id_event);
11     $reply = new ReplyReview;
12     $reply->id = Uuid::generate(4);
13     $reply->id_review = $request->id_review;
14     $reply->isi = $request->isi;
15     $reply->save();
16 }

```

Kode Sumber 4.39 Fungsi Event Mengelola Data Review Event

4.3.22 Implementasi Proses Pengguna Umum Menggunakan Fitur *Contact Us*

Pengguna umum dapat menggunakan fitur *Contact Us* dengan mengirimkan pesan yang diatur dalam Kode Sumber 4.40 dengan fungsi **addContactus()**. Pengguna umum yang belum terdaftar menjadi *member* akan memasukkan data *e-mail*, sedangkan pengguna yang sudah menjadi *member* akan otomatis diambil data **usernamenya**.

```

1  public function addContactus(Request $request)
2  {
3      $contactus = new Contactus;
4      if(Auth::check())
5      {
6          $contactus->id = Auth::user()->id;
7      }
8      $contactus->id_contactus = Uuid::generate(4);
9      $contactus->email = $request->email;
10     $contactus->judul = $request->judul;
11     $contactus->isi = $request->isi;
12     $contactus->status = 'NOT REPLIED';
13     $contactus->save();
14 }

```

Kode Sumber 4.40 Fungsi Pengguna Umum Menggunakan Fitur *Contact Us*

4.3.23 Implementasi Proses Pengguna Umum Menggunakan Fitur *Frequently Asked Questions*

Halaman *Frequently Asked Questions* dapat dilihat oleh pengguna dengan Kode Sumber 4.41 dan ditampilkan dengan fungsi `index()`.

```
1 public function index()
2 {
3     $this->data['faqs'] = Faq::get();
4 }
```

Kode Sumber 4.41 Fungsi Pengguna Umum Menggunakan Fitur *Frequently Asked Questions*

4.3.24 Implementasi Proses Event Mengisi *Review* untuk *Tenant*

Event dapat mengisi *review* untuk *tenant* yang mendaftar kepadanya. Daftar *tenant* dapat dilihat dengan fitur `tenant()` dan *event* dapat mengisi *review* untuk *tenant* dengan fitur `addreviewTenant()` seperti yang dijelaskan pada Kode Sumber 4.42.

```
1 public function tenant(Request $request)
2 {
3     $events = Event::with('request')->whereHas('request', function($q)
4 use ($request){
5     $q->where('status', 'ACCEPTED')->where('id_tenant', $request-
6 >id_tenant);
7     })
8     ->where('id', Auth::user()->id)
9     ->get();
10    $this->data['events'] = $events;
11    $this->data['tenant'] = Tenant::findOrFail($request->id_tenant);
12 }
13 public function addreviewTenant(Request $request)
14 {
15     $reviewtenant = new Review;
16     $reviewtenant->id_review = Uuid::generate(4);
17     $reviewtenant->id_tenant = $request->id_tenant;
18     $reviewtenant->id_event = $request->id_event;
19     $reviewtenant->email = $request->email;
```



```

20     $reviewtenant->sender = 'Event';
21     $reviewtenant->judul_review = $request->judul_review;
22     $reviewtenant->kualitas_produk_tenant = $request-
>kualitas_produk_tenant;
23     $reviewtenant->pelayanan_tenant = $request->pelayanan_tenant;
24     $reviewtenant->harga_tenant = $request->harga_tenant;
25     $reviewtenant->dekor_tenant = $request->dekor_tenant;
26     $reviewtenant->other_review = $request->other_review;
27     $reviewtenant->rating = ($request->kualitas_produk_tenant+$request-
>pelayanan_tenant+$request->harga_tenant+$request->dekor_tenant)/4;
28     $reviewtenant->save();
29 }

```

Kode Sumber 4.42 Implementasi Proses *Event* Mengisi *Review* untuk *Tenant*

4.3.25 Implementasi Proses *Tenant* Mengisi *Review* untuk *Event*
Tenant dapat mengisi *review* untuk *tenant* yang mendaftar kepadanya. Daftar *event* dapat dilihat dengan fitur **event()** dan *tenant* dapat mengisi *review* untuk *event* dengan fitur **addreviewEvent()** seperti yang dijelaskan pada Kode Sumber 4.43.

```

1  public function event(Request $request)
2  {
3      $tenants = Tenant::with('request')->whereHas('request', function($q)
4  use ($request){
5      $q->where('status', 'ACCEPTED')->where('id_event', $request-
>id_event);
6      })
7      ->where('id', Auth::user()->id)
8      ->get();
9      $this->data['event'] = Event::findOrFail($request->id_event);
10     $this->data['tenants'] = $tenants;
11 }
12 public function addreviewEvent(Request $request)
13 {
14     $reviewevent = new Review;
15     $reviewevent->id_review = Uuid::generate(4);
16     $reviewevent->id_tenant = $request->id_tenant;
17     $reviewevent->id_event = $request->id_event;
18     $reviewevent->sender = 'Tenant';
19     $reviewevent->judul_review = $request->judul_review;

```

20	\$reviewevent->ketepatan_waktu_event = \$request->ketepatan_waktu_event;
21	\$reviewevent->venue_event = \$request->venue_event;
22	\$reviewevent->hiburan_event = \$request->hiburan_event;
23	\$reviewevent->fasilitas_event = \$request->fasilitas_event;
24	\$reviewevent->other_review = \$request->other_review;
25	\$reviewevent->rating = (\$request->ketepatan_waktu_event+\$request->venue_event+\$request->hiburan_event+\$request->fasilitas_event)/4;
26	\$reviewevent->save();
27	}

Kode Sumber 4.43 Implementasi Proses *Tenant* Mengisi *Review* untuk *Event*

4.3.26 Implementasi *Pop-up* Pengguna Umum untuk Mendaftar sebagai *Event* atau *Tenant*

Proses mendaftar sebagai *event* atau *tenant* diatur oleh sistem dengan *pop-up* muncul pada halaman utama diatur dalam Kode Sumber 4.44. Fungsi ini akan mengarahkan pengguna ke proses pendaftaran.

1	<code>\$ (document).ready(function(){</code>
2	<code> setTimeout(function(){</code>
3	<code> \$('#myModal').modal('toggle');</code>
4	<code> },8000);</code>
5	<code>});</code>

Kode Sumber 4.44 Implementasi *Pop-up* Pengguna Umum Mendaftar sebagai *Event* atau *Tenant*

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dilakukan tahap uji coba dan evaluasi CallTenant. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas yang telah dijabarkan pada Bab III. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

- Laptop
Tipe : Sony Vaio E-Series
- Prosesor : Intel® Core™ i5-CPU (2.50 GHz)
- Memori/RAM : 8GB

Pengujian dilakukan menggunakan *browser* Google Chrome versi 63.0 dengan mengakses alamat calltenant.interakses.com.

5.2 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan pada CRM yang diterapkan dalam CallTenant. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas serta pengisian kuesioner meliputi ketertarikan responden pada CRM yang disediakan. Pengujian berfokus pada pengalaman responden menggunakan CallTenant yang diproyeksikan dalam kuesioner yang menilai ketertarikan responden terhadap CRM CallTenant.

5.2.1 Pengujian Fungsionalitas

Pengujian kebutuhan fungsional dilakukan secara mandiri dengan menggunakan metode *black box*. Metode *black box* adalah metode di mana pengujian ditekankan pada pola masukan dan keluaran yang sesuai dengan skenario pada Tabel 3.4 hingga Tabel

3.40. Pengujian yang dilakukan mengacu pada kasus penggunaan yang dijelaskan pada Bab III.

5.2.2 Pengujian Ketertarikan Responden terhadap CRM

Selain pengujian yang dilakukan untuk melihat kesesuaian masukan dengan keluaran lewat pengujian fungsional, dilakukan pengujian kepada pengguna dengan skenario tahap pengujian seperti pada Tabel 3.4 hingga Tabel 3.40 untuk mengetahui seberapa besar ketertarikan responden terhadap CRM. Pengujian dilakukan oleh 25 orang di mana sepuluh orang merupakan *event*, sepuluh orang merupakan *tenant*, dan lima orang akan berperan sebagai admin.

Setelah itu, responden mengisi kuesioner seperti pada Lampiran 1 hingga Lampiran 3 untuk menilai ketertarikan responden pada CRM. Penilaian kuesioner pada setiap pertanyaannya diproyeksikan dalam skor dengan skala 1-5 yaitu 1 berarti sangat tidak setuju, 2 berarti tidak setuju, 3 berarti netral, 4 berarti setuju, dan 5 berarti sangat setuju.

5.2.2.1 Demografi Responden

Responden yang terlibat dalam pengisian kuesioner dibagi menjadi 5 kategori, yaitu umur, pekerjaan, pendapatan rata-rata per bulan, pengalaman menjadi *event/tenant*, serta pernah menyelenggarakan/mengikuti *event* atau tidak. Data demografi responden menunjukkan bahwa responden terbanyak yaitu berusia 26-35 tahun sebanyak 13 responden (52%) pada Tabel 5.1. Pada umur tersebut, responden banyak yang sudah memulai bisnis dengan memiliki *tenant* atau menyelenggarakan *event* sesuai dengan kriteria yang dibutuhkan bahwa responden memiliki pekerjaan sebagai *event* ataupun *tenant*. Sedangkan, responden yang memiliki pekerjaan selain *event/tenant* dapat mewakili pengujian CRM sebagai admin. Data pada Tabel 5.4 menunjukkan bahwa sebagian besar responden berpenghasilan kurang lebih Rp3.000.000-Rp5.000.000 karena responden sebagian besar sudah terjun menjadi *tenant/event* selama lebih dari 3 tahun yang dijelaskan pada Tabel 5.3 dan sudah pernah

mengikuti atau menyelenggarakan *event* kurang lebih 5-10 kali sesuai dengan keterangan pada Tabel 5.5.

a. Umur Responden

Tabel 5.1 Distribusi Responden Berdasarkan Umur

Umur (tahun)	Frekuensi	Persentase (%)
16-25	9	36
26-35	13	52
36-45	3	12
Total	25	100

b. Pekerjaan Responden

Tabel 5.2 Distribusi Responden Berdasarkan Pekerjaan

Pekerjaan	Frekuensi	Persentase (%)
<i>Event</i>	10	40
<i>Tenant</i>	10	40
Mahasiswa/Pelajar	4	16
Lain-Lain	1	4
Total	25	100

c. Pengalaman Menjadi *Tenant/Event*

Tabel 5.3 Distribusi Responden Berdasarkan Pengalaman Menjadi *Event/Tenant*

Durasi (Tahun)	Frekuensi	Persentase (%)
<1	2	8
1-3	6	24
>3	12	48
Tidak pernah	5	20
Total	25	100

d. Pendapatan Rata-Rata per Bulan

Tabel 5.4 Distribusi Responden Berdasarkan Pendapatan Rata-Rata per Bulan

Pendapatan	Frekuensi	Persentase (%)
<Rp1.000.000	1	4
Rp1.000.000- <Rp3.000.000	4	16
Rp3.000.000- <Rp5.000.000	8	32
>Rp5.000.000	7	28
Tidak bekerja	5	20
Total	25	100

- e. Pernah Menyelenggarakan/Mengikuti *Event*

Tabel 5.5 Distribusi Responden Berdasarkan Jumlah *Event* yang Pernah Diikuti

Jumlah event yang pernah diikuti	Frekuensi	Persentase (%)
<2	2	8
2-5	5	20
5-10	10	40
>10	8	32
Total	25	100

5.2.2.2 Hasil Kuesioner

Berikut merupakan hasil akhir skor kuesioner yang diisi oleh responden berdasarkan pertanyaan yang dijelaskan pada

Tabel 5.6 hingga Tabel 5.8 di mana diproyeksikan dalam skor 1 yang berarti sangat tidak setuju, 2 berarti tidak setuju, 3 berarti netral, 4 berarti setuju, dan 5 berarti sangat setuju.

Hasil kuesioner ini dapat disimpulkan bahwa sebagian besar responden yang merupakan *tenant*, *event*, dan admin setuju bahwa fitur yang dibuat mempermudah *tenant* dalam melakukan aktivitas dalam CallTenant. Hal ini dapat dilihat pada Tabel 5.6 bahwa hasil skor tertinggi dan terbanyak 70% memberikan skor 4 (setuju) pada fitur *contact us*, 50% memberikan skor 5 (sangat setuju) pada fitur FAQ, 60% memberikan skor 4 (setuju) pada fitur *feedback*, 50%

memberikan skor 5 (sangat setuju) pada fitur *reward*, 50% memberikan skor 5 (sangat setuju), 50% memberikan skor 5 (sangat setuju) pada fitur *wallet*, dan 70% memberikan skor 4 (setuju) pada fitur *review*.

Hasil skor tertinggi dan terbanyak pada Tabel 5.7 dijelaskan bahwa 60% memberikan skor 4 (setuju) pada fitur *contact us*, 50% memberikan skor 4 (setuju) pada fitur FAQ, 60% memberikan skor 4 (setuju) pada fitur *feedback*, 40% memberikan skor 5 (sangat setuju) pada fitur *reward*, 70% memberikan skor 4 (setuju) pada fitur *wallet*, dan 50% memberikan skor 5 (sangat setuju) pada fitur *review*.

Sedangkan hasil skor tertinggi dan terbanyak pada Tabel 5.8 dijelaskan bahwa 60% memberikan skor 4 (setuju) pada fitur laporan FAQ, 80% memberikan skor 4 (setuju) pada fitur laporan keuangan, 60% memberikan skor 4 (setuju) pada fitur *contact us*, 40% memberikan skor 5 (sangat setuju) pada fitur *reward* dan *point*, 40% memberikan skor 5 (sangat setuju) pada fitur laporan *user*, 40% memberikan skor 5 (sangat setuju) pada fitur laporan *tenant*, dan 40% memberikan skor 5 (sangat setuju) pada fitur laporan *event*, 50% memberikan skor 5 (sangat setuju) pada fitur laporan *feedback*, serta 50% memberikan skor 4 (setuju) pada fitur laporan *review*.

Tabel 5.6 Hasil Kuesioner Responden *Tenant*

No	Pertanyaan	Skor				
		1	2	3	4	5
1	Tingkat kemudahan dan kelengkapan mencari informasi dalam menggunakan fitur <i>Contact Us</i>	0%	0%	0%	70%	30%
2	Tingkat kemudahan dan kelengkapan mendapatkan informasi dalam menggunakan fitur <i>Frequently Asked Questions</i>	0%	0%	0%	50%	50%

3	Tingkat ketertarikan terhadap CallTenant atas adanya fitur <i>Feedback</i>	0%	0%	0%	60%	40%
4	Apakah anda tetap ingin menggunakan CallTenant dengan adanya fitur <i>Point</i> dan tukar dengan <i>Reward</i> ?	0%	0%	10%	40%	50%
5	Apakah jenis <i>reward</i> yang ditawarkan menarik bagi anda?	0%	0%	30%	30%	40%
6	Apakah hadiah <i>point</i> (dari pertama kali pendaftaran sebagai <i>member</i> dan mengisi <i>feedback</i>) menarik bagi anda?	0%	0%	10%	40%	50%
7	Apakah fitur <i>wallet</i> meningkatkan kualitas transaksi anda?	0%	0%	10%	40%	50%
8	Apakah fitur <i>review</i> mempermudah anda dalam mengelola informasi?	0%	0%	0%	70%	30%

Tabel 5.7 Hasil Kuesioner Responden *Event*

No	Pertanyaan	Skor				
		1	2	3	4	5
1	Tingkat kemudahan dan kelengkapan mencari informasi dalam menggunakan fitur <i>Contact Us</i>	0%	10%	0%	60%	30%
2	Tingkat kemudahan dan kelengkapan mendapatkan informasi dalam menggunakan fitur <i>Frequently Asked Questions</i>	0%	0%	20%	50%	30%

3	Tingkat ketertarikan terhadap CallTenant atas adanya fitur <i>Feedback</i>	0%	0%	0%	60%	40%
4	Apakah anda tetap ingin menggunakan CallTenant dengan adanya fitur <i>Point</i> dan tukar dengan <i>Reward</i> ?	0%	0%	10%	60%	30%
5	Apakah jenis <i>reward</i> yang ditawarkan menarik bagi anda?	0%	0%	20%	40%	40%
6	Apakah hadiah <i>point</i> (dari pertama kali pendaftaran sebagai <i>member</i> dan mengisi <i>feedback</i>) menarik bagi anda?	0%	0%	20%	60%	20%
7	Apakah fitur <i>wallet</i> meningkatkan kualitas transaksi anda?	0%	0%	0%	70%	30%
8	Apakah fitur <i>review</i> mempermudah anda dalam mengelola informasi?	0%	0%	0%	50%	50%

Tabel 5.8 Hasil Kuesioner Responden Admin

No	Pertanyaan	Skor				
		1	2	3	4	5
1	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Frequently Asked Questions</i> (Melihat, Menambah, Mengubah, dan Menghapus data)	0%	0%	20%	60%	20%
2	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Keuangan (<i>Wallet</i> dan <i>Payment</i>)	0%	0%	0%	80%	20%

3	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Contact Us</i> (Melihat dan Membalas pesan)	0%	0%	0%	60%	40%
4	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Reward</i> dan <i>Point</i> (Melihat, Menambah, Mengubah, dan Menghapus data)	0%	0%	20%	40%	40%
5	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>User</i>	0%	20%	0%	40%	40%
6	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Event</i>	0%	0%	20%	40%	40%
7	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Tenant</i>	0%	0%	20%	40%	40%
8	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Feedback</i>	0%	0%	0%	40%	60%
9	Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Review</i> (<i>Tenant</i> dan <i>Event</i>)	0%	0%	0%	60%	40%

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi pengujian ketertarikan responden terhadap CRM.

5.3.1 Evaluasi Pengujian Fungsionalitas

Hasil pengujian fungsionalitas secara keseluruhan dapat dilihat pada Tabel 5.9. Data berdasarkan tabel tersebut menunjukkan bahwa semua skenario pengujian berhasil dan program berjalan dengan baik sehingga pada Tabel 5.9 dapat ditarik kesimpulan bahwa fungsionalitas dari aplikasi bekerja sesuai dengan yang diharapkan.

Tabel 5.9 Evaluasi Pengujian Fungsionalitas

Kode Kasus Penggunaan	Kasus Penggunaan	Hasil
SP-001	Mengelola data <i>frequently asked questions</i>	Berhasil
SP -002	Mengelola pelaporan keuangan	Berhasil
SP -003	Membalas pesan <i>contact us</i>	Berhasil
SP -004	Mengelola data <i>reward</i> dan <i>point</i>	Berhasil
SP -005	Mengelola pelaporan data <i>tenant</i>	Berhasil
SP -006	Mengelola pelaporan data <i>event</i>	Berhasil
SP -007	Mengelola pelaporan data user	Berhasil
SP -008	Melihat hasil <i>feedback</i>	Berhasil
SP -009	Menggunakan <i>reward event</i>	Berhasil
SP -010	Menggunakan <i>reward tenant</i>	Berhasil
SP -011	<i>Event</i> mendapatkan <i>point</i>	Berhasil
SP -012	<i>Tenant</i> mendapatkan <i>point</i>	Berhasil
SP -013	Mengisi <i>feedback</i>	Berhasil
SP -014	Mengisi <i>wallet</i>	Berhasil
SP -015	Melihat riwayat konfirmasi <i>topup wallet</i>	Berhasil
SP -016	Menggunakan <i>reward event</i>	Berhasil
SP -017	Melihat riwayat pemakaian <i>point</i>	Berhasil
SP -018	Melihat riwayat pemakaian <i>wallet</i>	Berhasil
SP -019	Mengelola data <i>review tenant</i>	Berhasil
SP -020	Mengelola data <i>review event</i>	Berhasil
SP -021	Menggunakan fitur <i>contact us</i>	Berhasil

SP -022	Menggunakan fitur <i>frequently asked questions</i>	Berhasil
SP -023	Mengisi <i>review</i> untuk <i>tenant</i>	Berhasil
SP -024	Mengisi <i>review</i> untuk <i>event</i>	Berhasil
SP -025	Mendaftar sebagai <i>event/tenant</i>	Berhasil

5.3.2 Evaluasi Pengujian Ketertarikan Responden terhadap CRM

Hasil skor setiap pertanyaan pada kuesioner yang didapatkan diambil hasil pilihan skor terbanyaknya pada setiap pertanyaannya kemudian ditentukan apakah berhasil atau tidak melalui penghitungan *Confidence Interval*. *Confidence Interval* adalah rentang antara dua nilai di mana nilai suatu *Sample Mean* tepat berada di tengah-tengahnya. Nilai sebuah *confidence interval* dapat dinyatakan dengan kemungkinan (*probability*) berapa *sample* dalam 100 kali pengambilan *samples* nilai *population mean* sesungguhnya akan masuk dalam sebuah rentang *sample mean* [22]. *Confidence Interval* [1] menggunakan rumus berikut:

$$CI = \bar{x} \pm Z \frac{s}{\sqrt{n}} \quad (5.1)$$

Pada Persamaan 5.1 \bar{x} adalah rata-rata dari nilai kuesioner, Z adalah nilai yang didapat dari Tabel 5.10, s adalah standar deviasi, dan n adalah jumlah responden.

Tabel 5.10 Tabel Nilai Z

<i>Z</i>	
80%	1.282
85%	1.440
90%	1.645
95%	1.960
99%	2.576
99.5%	2.807
99.9%	3.291

Hasil penghitungan pengujian evaluasi menggunakan rumus *confidence interval* dapat dilihat pada Tabel 5.11 untuk *Tenant*, Tabel 5.12 untuk pihak *Event*, serta Tabel 5.13 untuk Admin. Penentuan

hasil didapat dari perbandingan nilai *error result* dan *error level* yang didapatkan pada penghitungan *error level* pada akurasi survei [23] pada masing-masing pertanyaan seperti yang tertera pada Lampiran 1 hingga Lampiran 3.

Apabila nilai *error result* di bawah persentase *error level* maka status menjadi Berhasil dan dianggap responden setuju dengan hasil survei yang telah diberikan. Sedangkan, apabila *error result* lebih dari persentase *error level* maka status menjadi Tidak Berhasil dan dianggap responden tidak setuju dengan hasil kuesioner yang telah diberikan. Kuesioner diberikan pada responden dengan populasi 25 orang dan kenyataan responden yang mengisi adalah 10 orang pada masing-masing *Event* dan *Tenant* serta 5 orang pada admin. Maka, dapat disimpulkan bahwa hasil evaluasi kuesioner pada Tabel 5.11 hingga Tabel 5.13 dinyatakan berhasil.

Tabel 5.11 Hasil Evaluasi Kuesioner Responden *Tenant*

Pertanyaan	Mean	Stdev	$Z^*(s/\sqrt{n})$	Confidence Interval			Error result	Error Level	Keterangan
				CI	Mean \uparrow	Mean \downarrow			
Tingkat kemudahan dan kelengkapan mencari informasi dalam menggunakan fitur <i>Contact Us</i>	4.3	0.48	0.30	0.30	4.60	4.00	3%	24%	Berhasil
Tingkat kemudahan dan kelengkapan mendapatkan informasi dalam menggunakan fitur <i>Frequently Asked Questions</i>	4.5	0.53	0.33	0.33	4.83	4.17	4%	24%	Berhasil
Tingkat ketertarikan terhadap Call/Tenant atas adanya fitur <i>Feedback</i>	4.4	0.52	0.32	0.32	4.72	4.08	4%	24%	Berhasil
Apakah anda tetap ingin menggunakan Call/Tenant dengan adanya fitur <i>Point</i> dan tukar dengan <i>Reward</i> ?	4.4	0.70	0.43	0.43	4.83	3.97	5%	24%	Berhasil
Apakah jenis <i>reward</i> yang ditawarkan menarik bagi anda?	4.1	0.88	0.54	0.54	4.64	3.56	6%	24%	Berhasil
Apakah hadiah <i>point</i> (dari pertama kali pendaftaran sebagai <i>member</i> dan mengisi <i>feedback</i>) menarik bagi anda?	4.4	0.70	0.43	0.43	4.83	3.97	5%	24%	Berhasil
Apakah fitur <i>wallet</i> meningkatkan kualitas transaksi anda?	4.4	0.70	0.43	0.43	4.83	3.97	5%	24%	Berhasil
Apakah fitur <i>review</i> mempermudah anda dalam mengelola informasi?	4.3	0.48	0.30	0.30	4.60	4.00	3%	24%	Berhasil

Tabel 5.12 Hasil Evaluasi Kuesioner Responden *Event*

Pertanyaan	Mean	Stdev	$Z^*(s/\sqrt{n})$	Confidence Interval			Error Result	Error Level	Keterangan
				CI	Mean ↑	Mean ↓			
Tingkat kemudahan dan kelengkapan mencari informasi dalam menggunakan fitur <i>Contact Us</i>	4.1	0.88	0.54	0.54	4.64	3.56	6%	24%	Berhasil
Tingkat kemudahan dan kelengkapan mendapatkan informasi dalam menggunakan fitur <i>Frequently Asked Questions</i>	4.1	0.74	0.46	0.46	4.56	3.64	5%	24%	Berhasil
Tingkat ketertarikan terhadap Call/Tenant atas adanya fitur <i>Feedback</i>	4.4	0.52	0.32	0.32	4.72	4.08	4%	24%	Berhasil
Apakah anda tetap ingin menggunakan Call/Tenant dengan adanya fitur <i>Point</i> dan tukar dengan <i>Reward</i> ?	4.2	0.63	0.39	0.39	4.59	3.81	4%	24%	Berhasil
Apakah jenis <i>reward</i> yang ditawarkan menarik bagi anda?	4.2	0.79	0.49	0.49	4.69	3.71	5%	24%	Berhasil
Apakah hadiah <i>point</i> (dari pertama kali pendaftaran sebagai <i>member</i> dan mengisi <i>feedback</i>) menarik bagi anda?	4	0.67	0.41	0.41	4.41	3.59	5%	24%	Berhasil
Apakah fitur <i>wallet</i> meningkatkan kualitas transaksi anda?	4.3	0.48	0.30	0.30	4.60	4.00	3%	24%	Berhasil
Apakah fitur <i>review</i> mempermudah anda dalam mengelola informasi?	4.5	0.53	0.33	0.33	4.83	4.17	4%	24%	Berhasil

Tabel 5.13 Hasil Evaluasi Kuesioner Responden Admin

Pertanyaan	Mean	Stdev	$Z^*(\sigma/\sqrt{n})$	Confidence Interval			Error result	Error Level	Hasil
				CI	Mean ↑	Mean ↓			
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Frequency Asked Questions</i> (Melihat, Menambah, Mengubah, dan Menghapus Data)	4	0.71	0.44	0.62	4.44	3.56	15%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Keuangan (<i>Waller</i> dan <i>Payment</i>)	4.2	0.45	0.28	0.39	4.48	3.92	10%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Contact Us</i> (Melihat dan Membalas Desain)	4.4	0.55	0.34	0.48	4.74	4.06	12%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data <i>Reward</i> dan <i>Point</i> (Melihat, Menambah, Mengubah, dan Menghapus Data)	4.2	0.84	0.52	0.73	4.72	3.68	18%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola Data <i>User</i>	4	1.22	0.76	1.07	4.76	3.24	27%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola Data <i>Event</i>	4.2	0.84	0.52	0.73	4.72	3.68	18%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola Data <i>Tenant</i>	4.2	0.84	0.52	0.73	4.72	3.68	18%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola Data <i>Feedback</i>	4.6	0.55	0.34	0.48	4.94	4.26	12%	39.20%	Berhasil
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola Data <i>Review</i> (<i>Tenant</i> dan <i>Event</i>)	4.4	0.55	0.34	0.48	4.74	4.06	12%	39.20%	Berhasil

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

5.4 Kesimpulan

Setelah melakukan perancangan, implementasi, uji coba, dan evaluasi, dapat diambil kesimpulan sebagai berikut:

1. CRM yang diimplementasikan ditujukan untuk CallTenant sesuai dengan kebutuhannya.
2. Fitur CRM yang dibuat pada CallTenant sudah sesuai dengan tahapan CRM yaitu *Acquire* (Menambah Pelanggan), *Enhance* (Meningkatkan Hubungan dengan Pelanggan), dan *Retain* (Mempertahankan Pelanggan) pada subbab 3.1.3.
3. Fitur *Wallet* yang dibuat bagi CRM CallTenant berguna untuk memudahkan proses transaksi pengguna sehingga menjadi daya tarik bagi CallTenant.
4. Evaluasi responden terhadap CRM dinilai menarik dan berhasil digunakan sesuai dengan hasil pada Tabel 5.11 hingga Tabel 5.13.

5.5 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem dimasa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Penambahan metode pembayaran supaya memudahkan pengguna melakukan transaksi dan menjadi daya tarik bagi CallTenant.
2. Pesan *Contact Us* yang dikirimkan oleh pengguna umum menggunakan email akan dikirimkan kembali balasannya ke *e-mail* si pengirim.

3. Penambahan fitur *withdrawal* yaitu penarikan uang dari *wallet*.
4. Penambahan kelengkapan data dalam halaman *Frequently Asked Questions*.

RBTC

DAFTAR PUSTAKA

- [1] M. A. Supriyanto, “Ini Perbedaan Generasi dan Kelebihannya,” 15 Mei 2016. [Daring]. Terdapat pada: <http://www.tribunnews.com/tribunners/2016/05/15/ini-perbedaan-generasi-dan-kelebihannya>. [Diakses 27 September 2017].
- [2] admin, “7 Bisnis Kreatif Yang Diminati Anak Muda,” 6 September 2016. [Daring]. Terdapat pada: <http://tabloidpeluangusaha.com/7-bisnis-kreatif-yang-diminati-anak-muda.html>. [Diakses 27 September 2017].
- [3] S. Montana dan M. M. Noor, “PENGEMBANGAN CUSTOMER RELATIONSHIP MANAGEMENT BERBASIS SISTEM E-COMMERCE,” *CommIT*, vol. 4, pp. 139-149, Oktober 2010.
- [4] M. Sebastian, “Custom CRM System written in PHP,MySQL,and jQuery,” [Daring]. Terdapat pada: <https://www.behance.net/gallery/15161191/Custom-CRM-System-written-in-PHP-MySQL-and-jQuery>. [Diakses 27 September 2017].
- [5] R. Chen, “PHP For Beginners: Building a Simple CRM from Scratch,” 1 March 2017. [Daring]. Terdapat pada: <https://css-tricks.com/php-beginners-building-simple-crm-scratch/>.
- [6] A. Meiriza, H. Novianti dan M. Temala, “PENERAPAN ELECTRONIC CUSTOMER RELATIONSHIP MANAGEMENT (E-CRM) DALAM PENJUALAN ALAT-ALAT ELEKTRONIK UNTUK MENINGKATKAN LAYANAN PELANGGAN PADA DUTA ELEKTRONIK PALEMBANG,” dalam *Konferensi Nasional Teknologi Informasi dan Aplikasinya*, Palembang, 2016.
- [7] “Pusat Bazaar,” Pusat Bazaar, [Daring]. Terdapat pada: <http://pusatbazaar.com>. [Diakses 19 Januari 2018].
- [8] “Zoho CRM Reviews: Overview, Pricing, and Features,” FinancesOnline, [Daring]. Terdapat pada: <https://reviews.financesonline.com/p/zoho-crm/>. [Diakses 29 December 2017].
- [9] A. Sulistio, “Analisa Usaha Event Organizer,” [Daring]. Terdapat pada: <http://www.analisausaha.com/analisa-usaha-event-organizer/>. [Diakses 27 September 2017].

- [10] E. Interior, “Pengertian Dan Penjelasan Tentang Booth/Pameran,” [Daring]. Terdapat pada: <http://www.interiorfurnituremurah.com/2016/01/pengertian-dan-penjelasan-tentang.html>. [Diakses 12 Januari 2018].
- [11] P. Kotler, *MARKETING MANAGEMENT* Prentice Hall Marketing, 11 penyunt., Prentice Hall, 2003.
- [12] A. Zuhdi dan H. Setiawan, “Implementasi Customer Relationship Management pada Perusahaan Kontraktor Pembangunan,” dalam *Konferensi Nasional Sistem Informasi*, Bandung, 2006.
- [13] F. Buttle, *Customer Relationship Management*, 1st penyunt., London: Routledge, 2004.
- [14] S. Srinivasan, *Cloud Computing Basics SpringerBriefs in Electrical and Computer Engineering*, Springer, 2014, pp. 19-22.
- [15] J. H. Fleming, Ph.D dan J. Asplund, *Human Sigma: Managing the Employee-Customer Encounter*, Simon and Schuster, 2007.
- [16] T. Musanto, “Faktor-Faktor Kepuasan Pelanggan dan Loyalitas Pelanggan: Studi Kasus pada CV. Sarana Media Advertising Surabaya,” *Jurnal Manajemen & Kewirausahaan*, vol. 6, pp. 123-136, 2004.
- [17] J. Griffin, *Customer Loyalty: How to Earn It, How to Keep It*, Jakarta: Erlangga, 2005.
- [18] P. Kotler dan G. Armstrong, *Dasar-Dasar Pemasaran*, Jakarta: Prenhallindo, 2005.
- [19] P. Timm, *Seven Power Strategies for Building Customer Loyalty*, Amacom, 2001.
- [20] N. Herucahyono, “Fitur Baru: BukaDompot – Dompot Virtual Anda,” BukaLapak, [Daring]. Terdapat pada: <https://blog.bukalapak.com/2013/09/fitur-baru-bukadompot-dompot-virtual-anda/>. [Diakses 12 Januari 2018].
- [21] “Kekurangan dan kelebihan dari sistem pembayaran elektronik,” unichange, [Daring]. Terdapat pada: https://unichange.me/in/articles/advantages_of_electronic_payment_systems. [Diakses 14 Januari 2018].
- [22] “SURVEY STATISTICAL CONFIDENCE: HOW MANY IS ENOUGH?,” GreatBrook, [Daring]. Terdapat pada:

<https://greatbrook.com/survey-statistical-confidence-how-many-is-enough/>. [Diakses Januari 2018].

- [23] “Survey Random Sample Calculator,” custom insight, [Daring]. Terdapat pada: <https://www.custominsight.com/articles/random-sample-calculator.asp>. [Diakses 6 Januari 2018].
- [24] “SURVEY STATISTICAL CONFIDENCE: HOW MANY IS ENOUGH?,” GreatBrook, [Daring]. Terdapat pada: <https://greatbrook.com/survey-statistical-confidence-how-many-is-enough/>. [Diakses 7 Januari 2018].
- [25] “Survey Random Sample Calculator,” costum insight, 2018. [Daring]. Terdapat pada: <https://www.custominsight.com/articles/random-sample-calculator.asp>. [Diakses 7 Januari 2018].

(Halaman ini sengaja dikosongkan)

RBTC

LAMPIRAN

Kuesioner Tingkat Kelayakan CallTenant bagi Event

Terima kasih banyak atas kesediaannya menjadi responden dalam pengerjaan Tugas Akhir kami, sebelumnya perkenalkan kami Rahmatin Nadia dan Kania Amalia dari Departemen Informatika. Kami mohon kesediaannya untuk mengisi kuisioner dibawah ini. Sekali lagi kami ucapkan terima kasih banyak

Nama Event

SleepFest

Berapakah umur anda?

- ☒ 16-25 tahun
- ☐ 26-35 tahun
- ☐ >36 tahun

Apakah pekerjaan anda?

- ☐ Event
- ☐ Tenant
- ☒ Mahasiswa/Pelajar
- ☐ Lain-Lain

Berapa lama pernah menjadi Event?

- ☒ <1 tahun
- ☐ 1-3 tahun
- ☐ >3 tahun
- ☐ Tidak pernah

Berapakah pendapatan rata-rata anda setiap bulan?

- ☐ <Rp1.000.000
- ☒ Rp1.000.000-Rp3.000.000
- ☐ Rp3.000.000-Rp5.000.000
- ☐ >Rp5.000.000
- ☐ Tidak bekerja

Berapa kali anda pernah menyelenggarakan atau mengikuti event?

- ☐ <2
- ☒ 2-5
- ☐ 5-10
- ☐ >10

Apakah hadiah point (dari pertama kali pendaftaran sebagai member dan mengisi feedback) menarik bagi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur wallet meningkatkan kualitas transaksi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur review mempermudah anda dalam mengelola informasi?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah anda tetap ingin menggunakan CallTenant dengan adanya fitur Point dan tukar dengan Reward?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Sangat Setuju

Apakah hadiah point (dari pertama kali pendaftaran sebagai member dan mengisi feedback) menarik bagi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur wallet meningkatkan kualitas transaksi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur review mempermudah anda dalam mengelola informasi?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Lampiran 1 Kuesioner Event

Kuesioner Tingkat Kelayakan CallTenant bagi Tenant

Terima kasih banyak atas kesediaannya menjadi responden dalam pengerjaan Tugas Akhir kami, sebelumnya perkenalkan kami Rahmatin Nadia dan Kania Amalia dari Departemen Informatika. Kami mohon kesediaannya untuk mengisi kuisioner dibawah ini. Sekali lagi kami ucapkan terima kasih banyak

Nama Tenant

Risoles Mayo

Berapakah umur anda?

- ☒ 16-25 tahun
- ☐ 26-35 tahun
- ☐ >36 tahun

Apakah pekerjaan anda?

- ☐ Event
- ☒ Tenant
- ☐ Mahasiswa/Pelajar
- ☐ Lain-Lain

Berapa lama pernah menjadi Tenant?

- ☐ <1 tahun
- ☒ 1-3 tahun
- ☐ >3 tahun
- ☐ Tidak pernah

Berapakah pendapatan rata-rata anda setiap bulan?

- ☐ <Rp1.000.000
- ☒ Rp1.000.000-Rp3.000.000
- ☐ Rp3.000.000-Rp5.000.000
- ☐ >Rp5.000.000
- ☐ Tidak bekerja

Berapa kali anda pernah menyelenggarakan atau mengikuti event?

- ☐ <2
- ☒ 2-5
- ☐ 5-10
- ☐ >10

Tingkat kemudahan dan kelengkapan mencari informasi dalam menggunakan fitur Contact Us

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

Tingkat kemudahan dan kelengkapan mendapatkan informasi dalam menggunakan fitur Frequently Asked Questions

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik



Tingkat ketertarikan terhadap CallTenant atas adanya fitur Feedback

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

Apakah anda tetap ingin menggunakan CallTenant dengan adanya fitur Point dan tukar dengan Reward?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah jenis reward yang ditawarkan menarik bagi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah hadiah point (dari pertama kali pendaftaran sebagai member dan mengisi feedback) menarik bagi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur wallet meningkatkan kualitas transaksi anda?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Apakah fitur review mempermudah anda dalam mengelola informasi?

	1	2	3	4	5	
Sangat Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Setuju

Lampiran 2 Kuesioner Tenant

Kuesioner Tingkat Kelayakan Call Tenant bagi Admin

Terima kasih banyak atas kesediaannya menjadi responden dalam pengerjaan Tugas Akhir kami, sebelumnya perkenalkan kami Rahmatin Nadia dan Kania Amalia dari Departemen Informatika. Kami mohon kesediaannya untuk mengisi kuisisioner dibawah ini. Sekali lagi kami ucapkan terima kasih banyak

Berapakah umur anda?

- ☒ 16-25 tahun
- ☐ 26-35 tahun
- ☐ >36 tahun

Apakah pekerjaan anda?

- ☐ Event
- ☐ Tenant
- ☐ Mahasiswa/Pelajar
- ☒ Lain-Lain

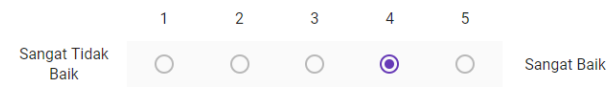
Berapa lama pernah menjadi Event?

- ☐ <1 tahun
- ☐ 1-3 tahun
- ☐ >3 tahun
- ☒ Tidak pernah

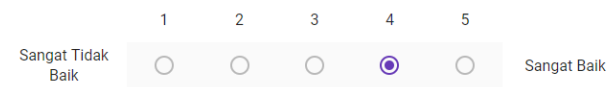
Berapakah pendapatan rata-rata anda setiap bulan?

- ☐ <Rp1.000.000
- ☒ Rp1.000.000-Rp3.000.000
- ☐ Rp3.000.000-Rp5.000.000
- ☐ >Rp5.000.000
- ☐ Tidak bekerja

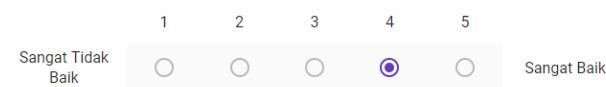
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Frequently Asked Question (Melihat, Menambah, Mengubah, dan Menghapus data)



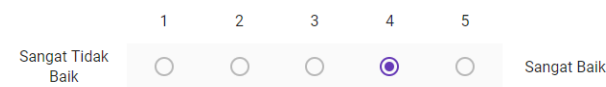
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Keuangan (Wallet dan Payment)



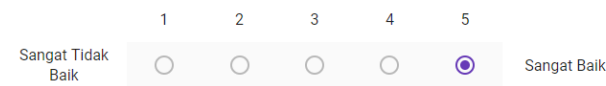
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Contact Us (Melihat dan Membalas pesan)



Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Reward dan Point (Melihat, Menambah, Mengubah, dan Menghapus data)



Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data User



Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Event

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Tenant

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Feedback

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

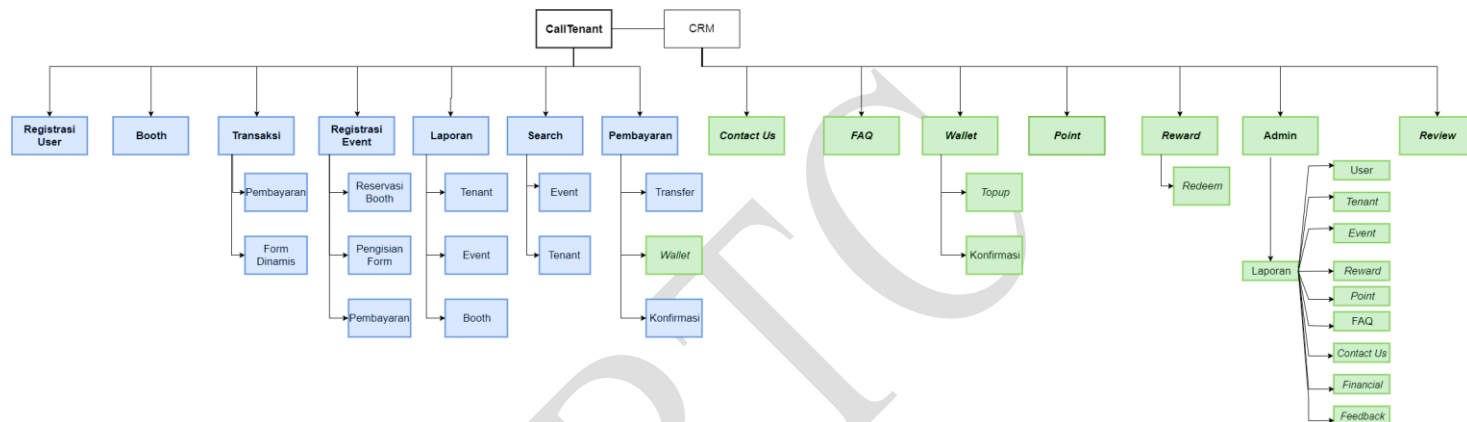
Tingkat Kemudahan, Efektivitas, dan Efisiensi dalam mengelola data Review (Tenant dan Event)

	1	2	3	4	5	
Sangat Tidak Baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Sangat Baik

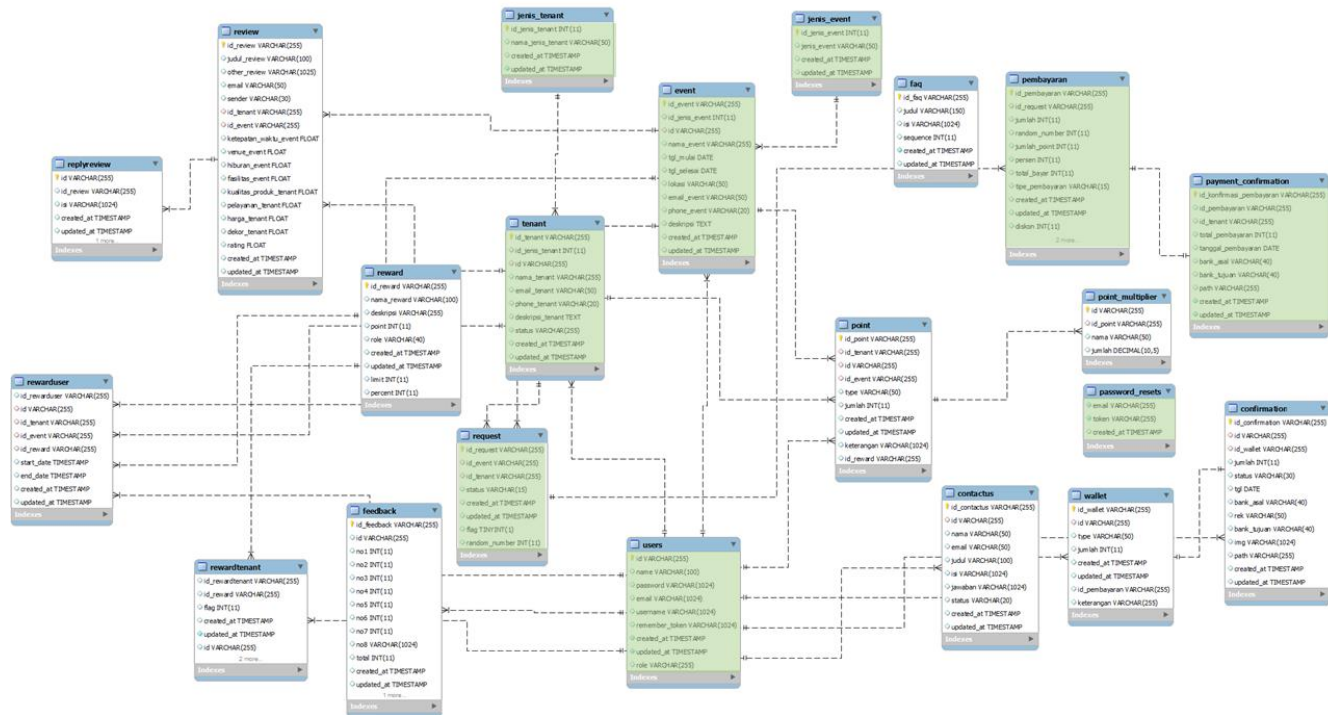
Lampiran 3 Kuesioner Admin

(Halaman ini sengaja dikosongkan)

RBTC



Lampiran 4 Perancangan Modul CRM pada CallTenant



Lampiran 5 Physical Data Model CRM CallTenant



BIODATA PENULIS

Penulis, **Kania Amalia**, lahir di Jakarta, 21 September 1996. Penulis adalah anak ketiga dari tiga bersaudara. Penulis menempuh pendidikan sekolah dasar di SD Pangudi Luhur Jakarta. Melanjutkan pendidikan sekolah menengah pertama di SMP Pangudi Luhur Jakarta dan penulis menempuh pendidikan menengah atas di

SMA Negeri 70 Jakarta. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi administrator Laboratorium Manajemen Informasi Teknik Informatika dan aktif dalam berbagai organisasi baik tingkat jurusan maupun fakultas.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi (MI) dan memiliki ketertarikan pada bidang *Web Application Development and Management*. Sebagai mahasiswa, penulis berperan aktif dalam beberapa organisasi kampus seperti staf ticketing pada acara Fantasi *closing* POMMITS 2015. Selain itu penulis juga menjadi staf Reeva pada SCHEMATICS 2015 dan menjadi BPH HUMAS SCHEMATICS 2016. Penulis dapat dihubungi melalui email: kaniaamalia21@gmail.com atau melalui line: kania21